OPTIMIZING SKYLINE QUERIES IN LARGE-SCALE UNCERTAIN GRAPH USING GRAPH NEURAL NETWORKS AND REINFORCEMENT LEARNING

BY

H M IKRAM KAYS

A thesis submitted in fulfillment of the requirement for the degree of Master in Computer Science and Information Technology.

Information & Communication Technology

International Islamic University Malaysia

DECEMBER 2025

ABSTRACT

Skyline queries play a critical role in multi-criteria decision-making by identifying data points that are not dominated by others, thus offering optimal choices to users. These queries are particularly valuable in transportation management, logistics, route optimization, and decision support systems. However, existing skyline query processing algorithms exhibit limited effectiveness when applied to large-scale and uncertain graph datasets, due to their reliance on exhaustive dominance comparisons, and sensitivity to uncertainty, which collectively result in incompatibility when applied in real-world graph environments. To address these challenges, this research proposes a novel skyline query processing framework that integrates Graph Neural Networks (GNNs) with Reinforcement Learning (RL), enabling effective representation learning over uncertain graph structures and adaptive skyline selection, a solution to ensure compatibility, and also test the potential scalability of the framework. As part of the contribution, large-scale uncertain graph datasets are systematically constructed with controlled size, density, and uncertainty levels to enable rigorous evaluation and scalability analysis. The proposed method is evaluated using 10-fold cross-validation, and performance is measured using accuracy, precision, recall, F1-score, and ROC-AUC. Experimental results demonstrate that while baseline skyline algorithms achieve acceptable accuracy, they suffer from significantly lower precision and recall, leading to suboptimal identification of skyline points. In contrast, the proposed GNN-RL framework achieves an accuracy of 98.97% alongside recall and F1-score above 98%, demonstrating strong robustness in uncertain graph settings. Furthermore, scalability experiments across varying dataset sizes confirm the suitability of the proposed approach for large-scale skyline query processing. This research contributes both theoretically and practically to intelligent data analytics and supports the United Nations Sustainable Development Goal (SDG) No. 9 which promotes resilient infrastructure, sustainable industrialization, and innovation through the development of scalable and intelligent data-driven technologies.

ملخص البحث

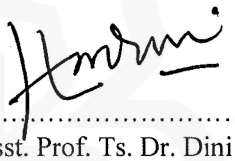
تلعب استعلامات الأفق (Skyline queries) دوراً حاسماً في اتخاذ القرارات متعددة المعايير من خلال تحديد نقاط البيانات التي لا تهيمن عليها نقاط أخرى، مما يوفر خيارات مثلى للمستخدمين. وتُعد هذه الاستعلامات ذات قيمة خاصة في إدارة النقل، والخدمات اللوجستية، وتحسين المسارات، وأنظمة دعم القرار. ومع ذلك، تُظهر خوارزميات معالجة استعلامات الأفق الحالية فعالية محدودة عند تطبيقها على مجموعات بيانات الرسوم البيانية الضخمة وغير المؤكدة، وذلك نظراً لاعتمادها على مقارنات الهيمنة الشاملة وحساسيتها تجاه عدم اليقين، مما يؤدي مجتمعاً إلى عدم التوافق عند تطبيقها في بيئات الرسوم البيانية الواقعية. ولمواجهة هذه التحديات، يقترح هذا البحث إطار عمل جديد لمعالجة استعلامات الأفق يدمج بين الشبكات العصبية للرسوم البيانية (GNNs) والتعلم المعزز (RL)، مما يتيح تعلم التمثيل الفعال عبر هياكل الرسوم البيانية غير المؤكدة والاختيار التكيفي لنقاط الأفق، وهو حل لضمان التوافق، واختبار قابلية التوسع المحتملة للإطار أيضاً. وكجزء من المساهمة، تم بناء مجموعات بيانات رسوم بيانية ضخمة وغير مؤكدة بشكل منهجي مع التحكم في الحجم، والكثافة، ومستويات عدم اليقين لتمكين التقييم الدقيق وتحليل قابلية التوسع. تم تقييم الطريقة المقترحة باستخدام التحقق المتبادل عشر مرات (10-fold cross-validation)، وتم قياس الأداء باستخدام الدقة، والإحكام، والاستدعاء، ودرجةF1 ، وROC-AUC . وتظهر النتائج التجريبية أنه بينما تحقق خوارزميات الأفق الأساسية دقة مقبولة، إلا أنها تعاني من انخفاض كبير في الإحكام والاستدعاء، مما يؤدي إلى تحديد دون المستوى الأمثل لنقاط الأفق. وفي المقابل، يحقق إطار عمل GNN-RL المقترح دقة تبلغ 98.97% إلى جانب معدلات استدعاء ودرجة F1 تزيد عن 98%، مما يثبت متانة قوية في بيئات الرسوم البيانية غير المؤكدة. بالإضافة إلى ذلك، تؤكد تجارب قابلية التوسع عبر أحجام بيانات متفاوتة ملاءمة النهج المقترح لمعالجة استعلامات الأفق واسعة النطاق. يساهم هذا البحث نظرياً وعملياً في تحليلات البيانات الذكية ويدعم الهدف رقم 9 من أهداف التنمية المستدامة للأمم المتحدة (SDG) الذي يعزز البنية التحتية المرنة، والتصنيع المستدام، والابتكار من خلال تطوير تقنيات ذكية وقابلة للتوسع تعتمد على البيانات.

APPROVAL PAGE

I certify that I have supervised and read this research and that in my opinion, it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a thesis for the degree of Master in Computer Science and Information Technology

.....

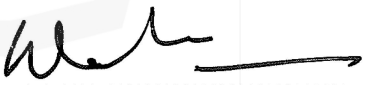
Asst. Prof. Dr. Raini Hassan
Supervisor



.....

Asst. Prof. Ts. Dr. Dini Oktarina Dwi Handayani
Co-Supervisor

I certify that I have read this research and that in my opinion it conforms to acceptable standards of scholarly presentation and is fully adequate, in scope and quality, as a thesis for the degree of Master in Computer Science and Information Technology



.....

Prof. Dr. Abdul Wahab Abdul Rahman
Internal Examiner

.....

Prof. Dr. Akram M Z M Khedher
Internal Examiner

.....

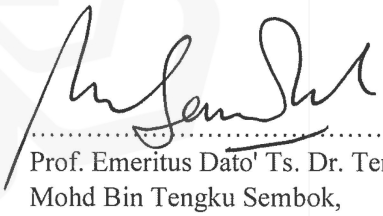
Assoc. Prof. Ts. Dr. Hitham Seddig Alhassan Alhussian
External Examiner

This thesis was submitted to the Department of Computer Science and is accepted as a fulfilment of the requirement for the degree of Master in Computer Science and Information Technology

.....

Asst. Prof. Dr. Azlin Binti Nordin
Head, Department of Computer Science

This thesis was submitted to the Kulliyyah of Information and Communication Technology and is accepted as a fulfillment of the requirement for the degree of Master in Computer Science and Information Technology



.....

Prof. Emeritus Dato' Ts. Dr. Tengku Mohd Bin Tengku Sembok,
Dean, Kulliyyah of Information and Communication Technology

DECLARATION

I hereby declare that this thesis is the result of my own investigations, except where otherwise stated. I also declare that it has not been previously or concurrently submitted as a whole for any other degrees at IIUM or other institutions.

H M IKRAM KAYS

Signature……Ikram kays…… Date……30/12/2025……

INTERNATIONAL ISLAMIC UNIVERSITY MALAYSIA

DECLARATION OF COPYRIGHT AND AFFIRMATION OF FAIR USE OF UNPUBLISHED RESEARCH

OPTIMIZING SKYLINE QUERIES IN LARGE-SCALE UNCERTAIN GRAPH USING GRAPH NEURAL NETWORKS AND REINFORCEMENT LEARNING

I declare that the copyright holder of this thesis is H M Ikram kays.

No part of this unpublished research may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise without prior written permission of the copyright holder except as provided below

1. Any material contained in or derived from this unpublished research may only be used by others in their writing with due acknowledgement.

2. IIUM or its library will have the right to make and transmit copies (print or electronic) for institutional and academic purpose.

3. The IIUM library will have the right to make, store in a retrieval system and supply copies of this unpublished research if requested by other universities and research libraries.

By signing this form, I acknowledged that I have read and understand the IIUM Intellectual Property Right and Commercialization policy.

Affirmed by H M Ikram kays

Ikram kays
…….
Signature

30/12/2025
……..
Date

ACKNOWLEDGEMENTS

First and foremost, I would like to express my deepest gratitude to Almighty Allah for His blessings and guidance throughout my journey in completing this master's degree. I am truly fortunate to have been supported by a remarkable group of people throughout my time at IIUM. As the Prophet Muhammad (peace be upon him) said:

"*Whosoever does not show gratitude to people has indeed not shown gratitude to Allah.*"

It is therefore with heartfelt appreciation that I thank my supervisor, Assistant Professor Dr. Raini Hassan, whose invaluable guidance, patience, and encouragement have been instrumental to the success of this research. I am also sincerely grateful to my co-supervisor, Assistant Professor Dr. Dini Oktarina Dwi Handayani, for her support and insightful advice throughout the process.

My sincere thanks go to the Faculty of KICT, especially the Deputy Dean, for their assistance during my challenging Viva-Voce session. His approval and my supervisor's recommendation played a key role in my ability to complete this endeavor.

A special mention to my beloved parents: Sarwar Hossain and Sultana Razia, who have always showered me with their unconditional love and prayers. They made sure that I never felt disheartened with their continuous encouragement and support. Despite their limited means, they did their best to meet my needs. Their hard work and enthusiasm kept me motivated throughout this journey. Even after being riddled with sickness, they never asked for anything for themselves, but instead wanted me to finish my studies for a better future. Their sacrifice will always be a part of this degree, and I am proud of that.

Special thanks to my brothers, Dr. H M Emrul Kays and Dr. H M Imran Kays, and my sister Sadia Momotaj Anonna for their unwavering support and guidance during this research phase. May Allah bless them abundantly in this world and the hereafter.

Finally, I wish to extend my sincere appreciation to the Ministry of Higher Education Malaysia for funding this research through the Fundamental Research Grant Scheme, Reference Code FRGS/1/2021/ICT01/UIAM/02/2. This generous support has made the completion of this thesis possible. To all who have helped me in any way, I extend my heartfelt thanks and offer prayers for your continued success and well-being.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

AP	Average Precision
AUPRC	Area Under the Precision Recall Curve
BBS	Branch and Bound Skyline
BN	Batch Normalization
CV	Cross Validation
DL	Deep Learning
ML	Machine Learning
DO	Dropout
F1-score	Harmonic mean of precision and recall
F1-drop	F1-score drop under noise / edge corruption (robustness indicator)
GD	Generational Distance
GNN	Graph Neural Network
RL	Reinforcement Learning
GNN+RL	Hybrid Graph Neural Network + Reinforcement Learning framework
HV	Hypervolume
nDCG@10	Normalized Discounted Cumulative Gain at 10
Precision@10	The proportion of the top 10 predicted skyline points that are true skyline members
Recall@10	The fraction of all true skyline points that appear within the top 10 predictions.
PSTV	Percentile skyline true value
Sp	Spacing

CHAPTER ONE

INTRODUCTION

1.1 Background

Today's data-driven decision-making environment requires efficient and accurate query resolution techniques that enable data to drive decisions efficiently and precisely. Skyline queries stand out as invaluable multi-criteria decision tool that extract non-dominated optimal solutions from datasets (Lind Mortensen, 2016); however, when applied to uncertain graph databases repositories with imprecision, ambiguity, or variability, existing skyline query processing methodologies fall short.

This research seeks to ease this complexity by incorporating deep learning algorithms, which are Graph Neural Networks (GNN) and Reinforcement Learning (RL), for processing skyline queries within uncertain graph databases. The primary goal is to develop an intelligent Framework capable of traversing graph structures without losing its effectiveness as an efficient and accurate means for pinpointing skyline points. At the core of this endeavor lies an aim to bridge the divide between existing skyline query processing methodologies and an ever-increasing need for robust uncertainty-aware techniques about graph databases (Abriola et al., 2023). By considering uncertain data elements and large graph structures as challenges for skyline query processing capabilities the researcher hopes to enable more precise, scalable, and insightful retrieval of information from skylines utilizing deep learning.

1.2 Problem Statement

The ever-increasing complexity and volume of data in modern databases present significant challenges for efficient and accurate query processing (Kim et al., 2012). In many real-world applications, such as transportation networks, disaster management, and supply chain systems, data is naturally represented as graphs and is inherently uncertain due to probabilistic relationships, incomplete observations, and dynamic conditions. These characteristics motivate the use of uncertain graph databases for skyline query processing. Although numerous skyline query algorithms have been proposed, most existing methods were originally designed for deterministic or tuple-based uncertain data and later extended to handle uncertainty. However, existing skyline query processing techniques are often ill-suited to handle the inherent uncertainties present in real-world data, particularly within uncertain graph databases (Lind Mortensen, 2016). In particular, they often fail to scale efficiently with increasing graph size and density, and they struggle to reliably identify true skyline points under probabilistic uncertainty and severe class imbalance.

Despite the rise of ML and DL, their integration into skyline query processing remains minimal, leaving significant room for optimization (Ahmed Mohamud et al., 2023). Existing skyline methods largely rely on dominance comparisons, pruning strategies, and probabilistic estimations, without exploiting the ability of DL models to capture complex structural dependencies and uncertainty patterns in large graphs. Consequently, there is a clear research gap in the development of a scalable, uncertainty-aware skyline query processing approach that leverages DL techniques to improve both effectiveness and robustness. Therefore, the problem addressed in this research is the absence of an effective and scalable skyline query processing framework capable of operating reliably on large-scale uncertain graph databases while preserving accurate skyline identification. Addressing this problem necessitates a systematic investigation of the limitations of existing baseline skyline algorithms, as well as an examination of how DL approaches can be leveraged to improve skyline query processing under uncertainty.

1.3 Research Questions

The research questions are as follows:

i. What are the limitations of baseline skyline query algorithms in addressing scalability and uncertainty for large-scale uncertain graph databases?

ii. How can deep learning be integrated into skyline query processing to improve scalability, accuracy, and uncertainty handling in large-scale uncertain graph databases?

1.4 Research Objective

The primary objective of this research is to propose a novel hybrid framework that leverages Graph Neural Networks (GNN) and Reinforced Learning (RL) to identify skyline queries in uncertain graph databases. This research also aims to explore the limitations of existing skyline query and develop an understanding of how these algorithms are unviable to overcome uncertainties in large graph datasets. The research objectives can be summarized as follows:

i. To investigate the capabilities of existing skyline query algorithms and their potential in large scale data.
ii. To propose a novel hybrid framework that incorporates GNN and RL to overcome scalability, accuracy and uncertainty challenges of skyline query processing in large scale graph databases.
iii. To empirically evaluate the performance of the proposed framework using measurable metrics such as accuracy, precision, recall, F1 score, ROC-AUC, multi-objective quality, robustness, efficiency and scalability.

This research aims to contribute a scalable and practical solution for skyline query processing, with potential applications in disaster management, intelligent transportation systems, and other domains that depend on efficient multi-criteria decision-making under uncertainty.

1.5 Significance of the Project

This research is significant in both academic and practical contexts as it addresses an important gap in skyline query processing for large-scale uncertain graph databases. From an academic perspective, the research contributes to the advancement of skyline query research by extending its scope beyond traditional deterministic and tuple-based uncertain data models to graph-structured environments characterized by probabilistic dependencies. By exploring deep learning approaches for skyline identification, this research provides new insights into how uncertainty and graph structure can be jointly addressed within skyline query processing, thereby enriching the theoretical understanding of enhanced uncertainty aware decision support queries.

From a methodological standpoint, the research offers a structured evaluation of deep learning-based skyline query processing in uncertain graphs. Through systematic experimentation and comparative analysis, the study demonstrates how alternative approaches can improve compatibility of skyline identification and possible scalability. The findings will provide a foundation for future research into adaptive and data-driven skyline query methods, encouraging further exploration of machine learning and deep learning optimization strategies in complex data environments.

The practical significance of this research lies in its applicability to real-world decision-support systems that utilize graph databases and operate under uncertainty. Many domains,

including disaster management, intelligent transportation systems, supply chain optimization, and recommendation systems, rely on timely and accurate multi-criteria decision-making using graph-structured data. By improving the reliability and scalability of skyline query processing in uncertain graph databases, this research supports more informed and effective decision-making in environments where uncertainty and data complexity are unavoidable.

1.6 Summary

This research aims to address the significant challenges faced by existing skyline query algorithms and pruning and filtration methods in large-scale uncertain graph databases. The research is motivated by the need for efficient and accurate query processing algorithms capable of handling the complexities and probabilistic nature of modern, large-scale data. Existing skyline query algorithms often fail to effectively identify skylines due to computational inefficiencies and scalability issues. Similarly, existing pruning and data filtration methods are impractical for uncertain graph databases, struggling to manage probabilistic variations and interdependencies. To overcome these limitations, this research will investigate the underlying reasons for the inadequacies of existing techniques and develop a novel hybrid framework using GNN and RL. The goal is to support more effective and reliable data-driven decision-making processes in complex and uncertain data environments, thereby addressing critical gaps in existing research and practice.

This research is organized into six chapters. Chapter one introduces the research background, problem statement, research questions, objectives, and significance of the study. Chapter two reviews existing literature on skyline query processing, uncertain data management, machine learning and deep learning approaches for graph-structured data. Chapter three presents the research methodology, including the proposed framework, data generation process, and experimental design. Chapter four reports the experimental results

and performance evaluation of the proposed approach in comparison with baseline methods. Chapter five provides a detailed discussion and analysis of the results, including robustness and scalability considerations. Finally, chapter six concludes the research by summarizing the key findings, outlining the main contributions, discussing limitations, and suggesting directions for future research.

CHAPTER TWO

LITERATURE REVIEW

2.1 Introduction

This chapter presents a comprehensive review of existing research related to skyline query processing, uncertain data management, machine learning and deep learning approaches for graph-structured data. The purpose of this chapter is to establish the theoretical foundation for the proposed research by examining prior studies, identifying prevailing techniques, and highlighting their limitations when applied to large-scale uncertain graph databases. By critically analyzing these reviews, this chapter aims to identify key research gaps related to scalability, uncertainty handling, and the limited integration of ML and DL methods in skyline query processing for uncertain graph databases.

2.2 Problem Context and Relevance

2.2.1 Background of Skyline Queries

2.2.1.1 Definition of Skyline Queries

Skyline queries are a fundamental concept in database systems that allows us to find a set of points or objects from the multidimensional dataset, that is not dominated by any other point (Ghosh et al., 2021). But in simpler terms it's a query asking to bring only those points from the dataset to which you can find maximum trade-off in every dimension and no other point can be strictly better in every dimension.

So particularly in terms of real-world decision making, where you have several criteria, but there's no one best option, skyline queries tend to really come into play. Take for example, that of a customer choosing a smartphone against price, battery life and camera quality. The phone might be cheaper but not have a good battery life or camera and vice versa; another phone might be more expensive but still come with more features. In this case the skyline queries help because we filter the phones that are dominated by the others, and we would end up with those phones which have tradeoffs but none of those are going to be inferior compared to the rest. From a mathematical perspective, let's assume that D is a dataset and each point in that dataset is a d-dimensional vector. Skyline of D denoted as Skyline (D), is defined as all points "$p \in D$" that do not have another point "$q \in D$" such that "q" dominates "p". Skyline queries are non-parametric in that there is no specific attribute ranking required prior to execution (Sorrentino, 2022).

A skyline query produces something which you can think of as a Pareto front, common in economics and multiobjective optimization, where there is no single point particularly good in all aspects (Razian et al., 2022). Therefore, the skyline set is a set of Pareto optimal points. For this reason, skyline queries in the database are a highly valuable query in applications when decision makers need to deal with multiple conflicting criteria without any need to explicitly rank or weight. In many real world problems, there are typically more than one objective that needs to be achieved simultaneously. These skyline queries enable efficient extraction of solutions that strike a good balance on various objectives. For example, a company may wish to minimize costs and delivery time in maximizing revenue earned on its products. The best solutions could be identified without having to convert the above objectives into a single optimization criterion through a skyline query. Despite their conceptual elegance, classical skyline query definitions implicitly assume deterministic attribute values, which limits their direct applicability in real-world scenarios where uncertainty and probabilistic dependencies are intrinsic.

Skyline queries provide a flexible and powerful tool to analyze multi ‘dimensional’ data where it is necessary to identify tradeoffs amongst many criteria. Across many domains from e-commerce to environmental monitoring, they provide a simple and powerful means to reduce large datasets to meaningful subsets of non-dominated points and thus are convenient for advanced decision support systems.

2.2.1.2 Importance and Applications of Skyline Queries

In recent years, skyline query has emerged as a cornerstone tool in modern data management systems, especially in applications with decisions for which different criteria are in conflict (Mohammed, 2024). These queries address the challenge of multi-criteria decision making (MCDM), a situation in which a single, optimal solution may not exist and users must instead assess different tradeoffs. Skyline queries provide an opportunity to enhance decision makers information by returning a set of non-dominated solutions which are more balanced in terms of the data available.

By far, skyline queries are one of the most important applications of them in recommendation systems, which employ the skyline query to compute appropriate personalized recommendations based on a set of attributes (Ke & Chang, 2019). In travel recommendation system for example, a skyline query may return hotels with a desirable combination of price, location and amenities. By enforcing a skyline query, none of the returned results are strictly worse than another in every dimension and thus users have the best possible trade offs without having to filter through a large number of options manually. This comes in handy on online shopping or e-commerce platforms where customers are recommended different products with different price quality and features (Kumar Sadineni, 2020). Platforms can provide recommendations for products that are suitable using skyline queries with respect to user preferences in multiple dimensions. In geographic information systems (GIS), skyline queries are also key; they are often used to find optimal location or

routing points that satisfy a number of spatial and non spatial criteria (Annisa & Angraeni, 2021). For instance, in the case of city planners, a skyline query can be used to choose the optimum place where to set up a new school accounting for all the possible different criteria such as distance from residential areas, the ease of access, the environment quality (Annisa et al., 2016).

Business intelligence and decision support systems also rely on skyline query (Loh et al., 2024). Skyline queries are used in these systems to assess the business options and compare various parameters, e.g., investment portfolios or supplier contracts regarding dimensions cost, risk and performance. For instance, a skyline query can be used by a company to determine suppliers who provide the mix of price, delivery time, reliability at the best cost. Without explicit ranking or weighting of criteria it also allows companies to focus on the most promising alternatives. Skyline queries are also used in environmental monitoring to identify critical points by selecting the best subsets of sensors from time series sensors in multiple dimensions (Khames et al., 2024). For example, in environmental science, one can use skyline queries to identify locations with the greatest threat of flooding, given the impact of rainfall, soil saturation etc., along with proximity to water bodies (R. Karanjit et al., 2023).

Additionally, skyline queries have become common Medical Decision Support System utilities being used by doctors and healthcare professionals to make treatment decisions on numerous patient factors (Zhang et al., 2021). One such example is that skyline queries can help select the best treatment option for an individual patient in terms of joint consideration of effectiveness, side effects, and cost. In personalized medicine this is particularly important, where treatment plans must be developed based on each patient's personal characteristics.

Skyline queries also play an important role in graph and network-based systems beyond structured data environments. As an example, skyline queries can help find influential users

on social networks in social network analysis by balancing such criteria as number of followers, engagement levels, and content quality (Zaman et al., 2015). Skyline queries are used in communication networks to determine 'optimal' routes to the transmission of data, including bandwidth, latency, and reliability (Kertiou et al., 2023). There are abundant use cases where skyline query proves its worth which makes research of this field very important and fruitful. However, many application-driven studies emphasize use cases without sufficiently addressing the computational implications of skyline queries when datasets grow large or become highly correlated. This gap becomes particularly pronounced in graph-based environments, where relational complexity further exacerbates skyline overpopulation and computational overhead.

2.2.1.3 Algorithms for Skyline Query Processing

Research on how to process skyline queries efficiently has been a main focus and has led to many algorithms for different kinds of datasets and applications. Block-Nested Loop (BNL) algorithm is one of the earliest and simplest algorithms for skyline query processing (Jeong et al., 2023). In BNL the possibility that one data point might dominate another is evaluated in a nested loop compared to every other data point (Borzsony et al., 2001). All such points are discarded if a point can be proven to be dominated. BNL is easier to implement and works nicely for small datasets; however, as the dataset size or dimensionality grows, BNL suffers from very poor performance because of all the pairwise comparisons. The time complexity of BNL is $O(n^2)$ where n is the number of data points, making it inefficient for large datasets (Endres et al., 2015).

To complement the BNL, the Divide & Conquer (D&C) algorithm was introduced. The improvement in performance brought by D&C recursively splitting the dataset into smaller subsets and computing the skyline of all subsets and then merging the results (Er-Rafyg et al., 2023). This is usually done along dimensions and skyline points for each

partition. D&C can reduce the number of comparisons in order to process the smaller subsets leading to the reduced number of skyline points, however this algorithm still suffers when working with high dimensional datasets where the number of skyline points is growing exponentially (Dehaki et al., 2020).

Progressive Skyline (PS) is another important algorithm for skyline query processing that returns skyline points incrementally as those are found (Papadias et al., 2005). Its main advantage lies in the fact that it can be very well used in environments where we do not need the entire skyline set, or where we require immediate responses (e.g., streaming data environments). In PS the data points are processed one at a time and skyline points are returned as soon as they form the skyline. That saves you from having to perform a bunch of unnecessary comparisons right away. The Branch and Bound Skyline (BBS) algorithm is very widely regarded as one of the most efficient algorithms for spatial and multi-dimensional datasets (Tiakas et al., 2015). The dataset is organized in an R-tree in order to facilitate pruning the search space using an R-tree data structure (Gao et al., 2015). The R-tree organizes data with points that are nearest each other and the algorithm has the ability to search only the portions of the dataset which are likely to contain skyline points. This considerably cuts down the amount of dominance comparisons needed, particularly for a large dataset. BBS is best suited to a small to medium dimensional dataset where the hierarchical structure of the R-tree allows for a very economical way to navigate the data. Although these algorithms demonstrate effectiveness in low-dimensional and static datasets, their performance degrades significantly as dimensionality increases, a phenomenon commonly referred to as the curse of dimensionality. Furthermore, most classical algorithms rely on fixed dominance checks and pruning rules, limiting their adaptability in dynamic or uncertain environments.

More specialized algorithms have been developed for high dimensional datasets, where the number of skyline points is likely to grow quickly. Among such algorithms is the Lattice Skyline algorithm that organizes the data in a lattice structure, facilitating exploration of the dominance among points more efficiently (Lakhal et al., 2017). Another

algorithm to high dimensional data is the Z-Sky algorithm, which uses a Z order curve to map multi-dimensional data point into one dimensional space for the ease of dominance comparison (Ken et al., 2009). Probabilistic skyline queries have been introduced for data points which carry probabilities or range of values in uncertain environments. These return queries were judged as likely non dominated. ProbSky is an algorithm for probabilistic skyline queries which optimizes this using advanced pruning techniques and probabilistic dominance rules to compute skyline sets in uncertain datasets efficiently (Kuo et al., 2022). The U-Skyline algorithm extends the existing skyline query algorithm in order to accommodate uncertainty in both the attributes of nodes as well as the relationships between them. U-Skyline provides a probabilistic model to calculate skyline dominance and to compute skyline points with respect to uncertain data (Liu et al., 2013). The efficient computation of Top K skyline objects also introduces an efficient algorithm for finding the top K skyline points in datasets with uncertain preferences. The key novelty of this algorithm lies in the aspect of how precisely it optimizes the selection of skyline points with user-specified criteria such that the memory complexity and run time is kept under control, given that in most cases users are only interested in a small fraction of the skyline. This algorithm uses probabilistic models to efficiently halve the search space, and it therefore becomes possible to quickly retrieve the most relevant skyline points (Sukhwani et al., 2021).

Each of these algorithms has a tradeoff between efficiency, scalability and complexity. D&C and BNL are simple and easy for users to interact with but cannot be used effectively with large datasets. PS is a good fit for streaming data, or any scenario in which you are required to produce incremental results. Spatial data are highly suitable for BBS, but BBS becomes less effective as dimensionality grows. At the same time, algorithms such as Lattice Skyline and Z-Sky, U-sky have been designed for the high-dimensionality setting, where conventional algorithms fail. With skyline queries continuing to be applied to increasingly more complex environments, particularly in the uncertain and distributed datasets, the area of new algorithm development is still an active research topic, where

focus is on improving their scalability, reducing the computation time, and better handling uncertainty.

2.2.2 Graph Databases

2.1.2.1 Definition and Overview of Graph Databases

Graph database is a NoSQL database that is designed to serve data in the form of graphs, where nodes are considered entities with relationships between them are represented through edges (Parmar & Roy, 2018). Properties or attributes can be placed on a node (or every node) and properties on an edge represent a relationship between two nodes. Graph databases work on infrastructure that is based on flexible, connected data model that is more suitable in applications with complicated relationships and network structure.

The fundamental components of a graph database are:

- Nodes: They represent individual entities or objects. For instance, nodes can represent people, in a social network, or locations, in a transportation network.
- Edges: They represent relationships or connections between nodes. In a social network, the edges are friendships or followers and in a transportation network, roads or routes.
- Attributes (or Properties): The attributes of both nodes and edges offer more information. For instance, a node representing yourself could have attributes of "Name" and "Age" etc.

Graph databases offer several advantages over relational databases when dealing with complex and interconnected datasets:

- Flexibility: A schema-less graph database doesn't require much redesign when data structures change. This flexibility is necessary for applications such as social

networks for which new types of relationships or entities may be introduced over time.

- Efficient Relationship Management: Since relationships are first-class in graph databases, the resulting query and analysis are far simpler than SQL joins, and much faster.
- Natural Representation of Networks: Many real world problems, such as recommendation systems, logistics and network analysis are very naturally represented as graphs. Intuitive graphs also come out as an easy entry point to model and query such systems.

Recently there has been a great adoption of graph databases by many businesses and industries as they began and recognized their ability to solve complex problem of relationship based queries (Robinson, 2015). Given the ability to deal with interconnected data, give fast query performance on highly relational data, and be easily adapted to changing data structures, graph databases are an indispensable piece of a modern data management systems (Lazarska & Siedlecka-Lamch, 2019).

2.1.2.2 Large-Scale Graph Database Definition and Challenges:

With the growth of scale in data, the scalability, query efficiency, and storage of data in large graph databases is becoming increasingly challenging. Scalability is one of the most important challenges in large scale graph databases (Sahu et al., 2017). Graph database systems are optimized for graphs that fit on a single machine and are small enough that they can fit in the memory. But, in large scale applications, datasets are much larger than the storage capacity of a single machine and distributed architectures are mandatory (Guyo & Hartmann, 2024). However, dividing a graph into smaller subgraphs in an optimal manner to minimize the number of edges crossing between different partitions to reduce communication overhead is hard and computationally expensive.

In large scale graph databases, query efficiency also becomes a problem. The more the graph grows, the longer it takes to traverse the graph looking for the data associated with it (Le-Phuoc et al., 2016). For instance, the task is computationally expensive for querying the shortest path of two nodes in a graph with millions of nodes when the graph is densely connected. Breadth-First Search (BFS), Depth-First Search (DFS) are often used efficient traversal algorithms, but even they fail to traverse large datasets (Shanker, 2024). This issue is overcome, to a certain extent, by indexing techniques such as graph partitioning and distributed indexing, available in graph databases, which helps to speed up query execution (Yang et al., 2012). Data storage is also a big challenge in large-scale graph databases. Graph databases are often optimized by compressing the graph by encoding the adjacency list or matrix in some way that reduces redundancy (Versari et al., 2020). For instance, edge collapsing compresses storage if edge connectivity patterns are similar. Furthermore, unlike relational databases that keep only small amounts of metadata with nodes and edges, graph databases often store large amounts of metadata along with the nodes and edges.

Concurrency control and transactional consistency are critical in large-scale graph databases, especially where the graph will be concurrently altered by multiple users or applications while querying the graph simultaneously. Many graph databases use eventual consistency models that allow updates to propagate through the system in an increasingly consistent manner for performance and scalability without the need for immediate consistency. Its implementation, however, brings problems in assuring the query result's accuracy and timeliness, especially in time sensitive systems such as recommendation engines or fraud detection systems.

Performance and storage issues in addition to graph analytics become increasingly complex at scale. Because performing analytical operations, such as calculating centrality measures (e.g. betweenness centrality, eigenvector centrality), detecting communities, or identifying

cliques in large graphs needs advanced algorithms that can efficiently handle large datasets (Kanavos et al., 2022). As the size and complexity of graphs used in applications like social media analysis, financial fraud detection, and biological network analysis grows, further improvements in graph partitioning, query optimization, and storage techniques are needed to continue to enable graph databases to be efficient.

2.1.2.3 Uncertainty in Graph Databases

In a graph database, we have uncertainty when edges or node or edge attributes aren't certain, or connectivity between nodes isn't absolute (Agarwal et al., 2020). In real world application, this uncertainty can be for example due to incomplete data, noisy measurements, probabilistic relationships or more generally due to not having enough information available. Uncertainty management is an important problem for graph databases in many applications, since the conclusions drawn from uncertainty are often misleading.

Probabilistic edges are one of the major sources of uncertainty in graph databases. In a probabilistic graph, each edge has an associated probability of existing (Hussain & Maab, 2021). As an example, consider a communication network where the edge between two nodes corresponds to a wireless connection between these nodes with a given probability caused by interference or signal loss. Likewise, with a social network we may have uncertain strength of a relationship between two individuals, resulting in probabilistic edges depicting the probability of interaction or influence. With probabilistic edges present in the graph, as well as on probabilistic nodes, specialized query processing techniques are needed to understand and manage these probabilistic edges.

Another common source of uncertainty with graph databases is with uncertain node attributes (Aggarwal et al., 2010). As an example, in biological networks the nodes are subject to measurement noise or incomplete data e.g. gene or protein expression levels. As well, in transportation networks, the travel time between two locations can vary because of traffic. In these cases, the nature of the graph can hardly be captured by existing deterministic queries, and probabilistic or fuzzy query models are required to deal with the uncertainty. These challenges have therefore led to development of probabilistic graph models. Into graph databases, these models extend, permitting probabilistic edges and uncertain node properties. What probabilistic graphical models like Bayesian networks or Markov Random Fields do is explicitly represent the uncertainties, so queries can be made to compute the likelihood of some paths, or the probability of a node in some cluster (Sucar, 2020). Probabilistic graphs are more complex in query processing than absolute graphs, because in general, query processing often involves computing the probabilities of a number of outcomes and summarizing those probabilities to obtain a meaningful result. To efficiently process queries in uncertain graph databases, techniques like Monte Carlo simulation (Emrich et al., 2012), sampling-based methods (Yuan et al., 2011) and probabilistic traversal algorithms (Aggarwal & Yu, 2009) have been developed. Fuzzy graph models represent another approach for handling uncertainty in graph databases (Ma & Yan, 2022). Fuzzy membership values are assigned in a fuzzy graph to edges and nodes, respectively expressing the degree to which an edge or node belongs to a certain set. Then fuzzy logic to query processing is applied, allowing for such more flexible or more nuanced queries that take uncertainty into account.

Uncertainty about graph databases is an important notion that needs to be taken into consideration before working with such database. To make graph databases more useful in real world applications, where data is often incomplete, noisy or probabilistic explicitly modeling uncertainty in edges and attributes can enable more accurate robust query processing.

2.1.3 Skyline Queries in Graph Databases

2.1.3.1 Current Use of Skyline Queries in Graph Databases

Skyline queries which were used to answer in multi criteria decision making, have also been applied to identifying optimal nodes or paths in terms of graph databases. Nodes in a graph database are frequently entities, such as people, places or products and the relationship between these nodes is represented by edges. In this context, skyline queries are useful for eliminating nodes (or paths) that best balance different criteria. For example, in the context of social network, skyline query could be used to retrieve the influential users that have a good balance of several followers and high engagement, and be related to a specific topic.

Graph databases typically use skyline queries in pathfinding and routing applications. As an example, in logistics or transportation networks, we have users who have to find the best paths between two nodes (locations, for example, two different cities), where “best” is comprised by several factors, including travel time, distance and road conditions (Gong et al., 2019). Such a skyline query returns all paths which are not dominated by others in terms of these attributes, so that the decision makers can pick one path. In systems where there is no one best path for all users and different users may value different points of interest, these queries prove to be particularly useful.

Skyline queries are also used for recommendation systems that use graph databases, where given nodes represent items and edges represent the relationship between users and items (Amin et al., 2020). Skyline queries help users find a set of the optimal items based on multiple attributes like price, popularity and rating. For example, time skyline queries can be used on an e-commerce graph database, to solve the product recommendations problem, that given some cost and quality and delivery times dimensions, you want to

recommend products to your customers such that none of the recommendations is worse than the other in every dimension.

Skyline query has also been observed to be used in network analysis to find core nodes or edges in a graph (Li et al., 2020). For instance, skyline queries are used in social networks to discover users that are popular in many dimensions at the same time, e.g., number of connections, interaction frequency, influence in certain topics or communities. Skyline queries are also used in communication networks where they identify the most reliable or fastest routes for data transmission balancing by factors such as bandwidth, latency and reliability (Khames et al., 2024).

Skyline queries on graph databases are unquestionably useful, however developing skyline queries for graph databases is challenging due to the complexity of graph structure and the necessity for efficient graph traversal algorithms.

2.1.3.2 Limitations of Skyline Queries in Graph Databases

Skyline queries have been useful in a variety of applications but their implementation in graph databases brings about a number of limitations. The computational complexity of skyline processing through a query in large and complex graphs is the most challenging component (Khan et al., 2012). This is because graph databases are very different to relational databases; the relationships between entities are equally as important as the entities themselves, and as a result, graph databases are inherently more complex. Graph traversal is required for evaluating dominance relationships among nodes or paths in a graph for multiple pairs of nodes or edges (Chen et al., 2021). The more nodes the graph has, the more comparisons which become exponentially difficult to scale. As this computational overhead makes skyline queries particularly difficult in large scale graphs,

such as those realizing transportation networks, social networks or biological networks that have millions of nodes and edges.

Another limitation is the difficulty of handling high-dimensional data in graph databases. Multiple attributes on nodes and edges in a graph are commonly present in most real-world applications, corresponding to independent dimensions in the skyline query. Consider a network, which can be transportation like here as each road (edge) has attributes such as distance, travel time, congestion level, and reliability. The curse of dimensionality applies in high dimensional datasets as the numbers of skyline points tend to grow exponentially. When the number of dimensions is increased, the probabilities that one point dominate the other decreases leading to a large number of skyline points. Processing skyline queries now significantly increases the computational cost due to the fact that each skyline point must be compared in multiple dimensions against all other skyline points. Now a days, real world graphs contain nodes and edges that may or may not exist. For example, in a transportation network, on the edge that represents the travel time between two locations, the travel time will be uncertain, based on traffic or weather conditions. skyline query algorithms are incapable of handling such uncertainties in data (Yong et al., 2014). This issue is addressed through probabilistic skyline queries, but such queries are computationally expensive and remain an active research area.

Finally, traversal and pathfinding are inherently more complex in graph databases, and in particular finding paths between any two nodes is inherently more complex when skyline queries involve finding optimal paths from any first node to any other second node. Now consider a typical skyline query where we want to find non-dominated nodes (or paths) based on some attributes but in a graph database usually we need to expend much effort to traverse the graph, which requires long time and consumes a lot of resources especially when it is large and densely connected. Existing skyline algorithms are not well-optimized for such graph traversal operations, making them less efficient when applied to graph databases.

2.1.3.3 Preferred Algorithms for Skyline Query Processing

Several algorithms were developed and adapted for skyline query in graph databases, with its own strength and weakness. Among its most popular techniques used is the Branch and Bound Skyline (BBS) algorithm, generally considered to be one of the most efficient algorithms with respect to skyline query processing (Papadias et al., 2003). The R-tree, or similar hierarchical index structure, is also used by BBS, which is able to efficiently prune non relevant nodes or edges to avoid considering irrelevant material. BBS is advantageous because it can reduce comparisons. It focuses only on regions of the graph that are promising, but its performance degrades in high dimensions and with uncertain attributes.

More recently, the algorithms for skyline query processing have seen great advances achieving unprecedented effectiveness in processing uncertain data and big scale databases. A lot of the work had been done using existing techniques such as Top-K, ProSky and U-Skyline but more and more work around solving existing limitations of these algorithms specifically in the areas of scalability and uncertainty management are being done.

Perhaps, the most notable algorithm is the U-Skyline algorithm which aims at finding skyline queries in uncertain databases. Compared with the existing skyline computation, this algorithm is geared to aggregate probability of tuples rather than per tuple. Collaboratively maximizing probability of being in the skyline, the U-skyline algorithm improves upon existing probabilistic models (Liu et al., 2013) . In particular, this algorithm is advantageous for real-world applications in which data uncertainty is ubiquitous.

Top-K Probabilistic Skyline Queries on Uncertain Data algorithm is also a preferred model for large and uncertain datasets. It aims at the problem of retrieving the top-K skyline

objects when the data is uncertain but probabilistic. This algorithm demonstrates that with a probabilistic dominance model, the search space can be reduced, and the retrieval process can be highly optimized so that only those skyline points that are most relevant are taken into consideration. Large datasets are handled efficiently by this algorithm, thus making the algorithm suitable for practical use where data uncertainty is ubiquitous (Yang et al., 2018).

Another notable algorithm is the ProbSky algorithm, which provides efficient skyline query computation in the case of probabilistic skyline queries over distributed data. For this algorithm they use a MapReduce based algorithm to quickly process large, high dimensional datasets while keeping the result accurate. Innovative pruning techniques are also used by ProbSky to reduce the search space to the point where skyline queries can be rapidly evaluated. (Kuo et al., 2022). In addition, the Stochastic Skyline algorithm extends skyline query processing further by incorporating the expected utility principle. The goal was for it to retrieve uncertain objects given their skyline probabilities under monotonic utility functions while accommodating the user's preferences. (Zhang et al., 2012). These developments reflect an ongoing evolution of skyline query algorithms towards more sophisticated algorithms which effectively deal with uncertainty and optimize the performance on large scale graph databases.

2.2 Knowledge Base and Theoretical Rigor

2.2.1 Theoretical Foundations of Skyline Queries

2.2.1.1 Deep Learning Algorithms for Skyline Queries

With large scale and uncertain graph databases, deep learning (DL) has been shown to be a powerful solution (Negro, 2021). As data size and complexity grows, existing skyline

query algorithms become computationally intensive, and innovative algorithms which leverage deep learning to optimize decision processes are required.

Data pruning and reduction are one of the key contributions of deep learning in skyline query processing. Instead of having to look at every data point in a dataset that can be very large, you can train these deep learning models to tell you what's irrelevant or redundant data points are, and sort them out before you run skyline queries. This greatly minimizes computational overheads. Furthermore, skyline queries can be supported to estimate dominance relationships using deep learning. Direct comparison is one form of existing processing, which becomes very difficult when compared to complex, high dimensional or uncertain datasets. The prediction of dominance relationship between two elements can be trained by deep learning models such as regression and neural networks, which lead to lower exhaustive pairwise comparison and other execution time requirement. Another important area is where deep learning excels is in handling uncertainty. In uncertain datasets where the attributes of nodes or edges are not deterministic, existing skyline algorithms perform poorly. In addition, DL improves the search for query execution plan. Typical skyline queries involve computing the most efficient path(s) between nodes on the basis of multi-dimensional attributes. In dynamic and distributed graph databases, where existing algorithms may have difficulty adapting to change quickly, reinforce learning (RL) algorithms, such as Q learning and policy gradient algorithms are useful in learning to optimize these paths over time (Chen & Chen, 2022).

In summary, DL is a great solution for skyline query processing, addressing issues with uncertainty handling, nonlinear relationships, query plan optimization and its scalability in large dataset. These offer compelling reasoning for incorporating deep learning techniques to boost skyline query processing in situations that are complex, large scale and uncertain.

2.2.1.2 Graph Neural Networks (GNN) for Skyline Queries

A Graph Neural Network (GNN) is a deep learning model that is specialized for structured data represented by a graph (Khemani et al., 2024). In contrast to machine learning models that operate over fixed size inputs, GNN naturally operate over large and complex graph inputs. Therefore, GNN is quite suitable for real world data where the essence of the system behavior depends heavily on entity-to-entity relationship. There are many domains in which graphs occur ubiquitously. For example in social networks the nodes (users) are connected by friendships or followers (edges). Such data is relational which is ideal for GNN's analysis. The key is that GNN is able to learn both the local and global information present in a graph, and therefore, be able to learn the attributes of each node as well as the influence of nearby nodes as well as the topology of the graph as a whole (Ying et al., 2019).

A GNN essentially convolutes their graph by aggregating information from neighbors to update its own feature representation. This way a GNN learns node representations (or embeddings) that also consist of how the node's neighbors are characterized. Repeating that process over multiple layers of the network enables nodes to build information based of nodes several hops away. GNN shows promise for problems such as node classification, link prediction, and graph level classification by exploiting dependencies between nodes through multiple layers.

The local and global structural information captured by GNN is important to analyze large and complex graph datasets. Although GNN is promising, the problem here is that we need to carefully consider several aspects when measuring GNN performance such as accuracy, scalability, computational efficiency, and robustness. Accuracy is one of the most important metrics to evaluate the performance of a model. Tasks on predicting the label of a node based on its attributes and that of its neighbors are still the most prominent tasks that GNN has proven superior accuracy in. For instance, GNN can precisely classify

users in a social network from their structure in terms of connections and interactions with other users. Scalability is also a crucial factor to GNN being applied to the large scale graph databases. It is often the case that graphs that arise in real applications involve just millions and billions of nodes and edges. Therefore, training GNN model can become computationally expensive, as the nodes must aggregate information of neighboring nodes. But despite that, several scalable GNN architectures have been developed already. For example, GraphSAGE uses sampling to aggregate info from a fixed size subset of neighbors, instead of operating on the whole graph (Hamilton et al., 2017). It reduces computational burden and enables GNN to scale much larger graphs.

The other important metric that is important is the computational efficiency, which is defined as the time and resource needed to train and deploy GNN on the large scale graphs. The message passing operations used in GNN are costly; however there are several optimization techniques to improve the computational efficiency of GNN. By mini batch training, GNN gradually process small subset of graph at a time and reduces the memory footprint and makes the training faster (Lin et al., 2020). The last but not least, the performance of GNN should be robust. The graph data usually contains noise or incompleteness and GNN needs to be immune to these noises without considerable loss of performance. It has been shown by several studies that GNN is relatively robust to noise and missing data for the case when enough information is known compared to existing graph algorithms, which require complete and accurate data (Fox & Rajamanickam, 2019). This robustness, in particular, makes GNN very attractive for real-world applications, where data quality is often poor.

However, skyline queries are difficult in graph databases, especially for large scale real world data because they are computationally complex, have high dimensionality, and uncertainty is present. But GNN has the potential to provide a strong solution to these challenges due to its ability to capture complex patterns and operate on graph structured data efficiently. The computational complexity of computing dominance relationships between nodes is one of the most important challenges in skyline query processing. Skyline

algorithms based on the existing algorithm rely on pairwise comparisons in order to decide that one node dominates another (Lee et al., 2016). This exponential growth of the number of comparisons renders skyline query processing computationally expensive, as the graph grows in size. With GNN, we can offset this problem via learning patterns of dominance relationships from the graph structure. Using the graph, the model first trains a GNN to predict whether a node is likely to dominate another without actually running the pairwise comparisons (Ahmed et al., 2021). This algorithm reduces computational burden and accelerates the skyline query.

GNN can be leveraged to account for uncertainty in such graph databases with probabilistic node attributes or edge weights compared to existing skyline query algorithms. When we do not have fixed values for node attributes or edge weights but rather distributions over them the uncertainty generates. For example, in a transportation network, the travel weight between two locations may be uncertain, due to uncertain edge weights. GNN can be used to learn from probabilistic distributions of node attributes such that they can predict the likelihood of dominance relationships in non-trivial environments. This is an important reason to why GNN is especially well suited to probabilistic skyline queries, where we wish to identify nodes likely to be a part of the skyline due to their probability of dominance. Combining uncertainty with the learning process will enable GNN to provide more accurate and reliable skyline query result in uncertain graph databases.

2.2.1.3 Reinforcement Learning for Skyline Queries

Reinforcement learning (RL) is a subset of deep learning concerned with how agents can learn good policies by interacting with their environment (Singh et al., 2021). An agent makes decisions at each step by choosing actions in the context of a given environment which takes one or more actions in return and gives them reward or penalty feedback. The agent's goal is to learn a strategy (or policy), or something that corresponds to applying a

function to the world that maximizes its cumulative reward over time. The difference is that in Supervised learning, the model learns from a labeled dataset but with RL the model learns by interaction and feedback. The agent explores the environment and learns from the consequences of its actions; gradually establishing better strategy, by learning which actions create positive outcome, and which need to be avoided. Often a Markov Decision Process (MDP) is given of the kind of environment, where we have a set of states together with a set of actions, rewards as to how the system moves on the basis that the agent decides, and transitions between states (Wachi & Sui, 2020).

Decision making in dynamic and uncertain environments is well suited for reinforcement learning. In many real-world situations data systems keep changing and what was the best decision in one point is not optimal in another due to changes in conditions. This uncertainty is handled effectively by the fact that RL's ability to continuously update its policy in response to new experiences. For example in robotics, RL has been used so that the robots can learn to navigate an environment using interaction with objects and adjusting movements based on signals from sensors. Due to reinforcement learning's adaptiveness to uncertainty environments, it is highly powerful. However, data systems usually operate in environments where action has out-of-deterministic outcomes. For example, network routing is uncertain on the availability and reliability of links because of the congestion or failure of the links. Here, RL has to learn how to balance rewards that are immediate (such as traveling down a fast route), with those that are forthcoming (for example, steering clear of congested streets). The agent can learn from both success and failure and comes up with its strategy based on the certainty of the environment.

Finally, reinforcement learning is a useful tool for optimizing decisions in dynamic complex environments where decisions must be realigned continuously (Piray & Daw, 2021). This makes RL a particularly well-suited application like network management, recommendation systems, and resource allocation in data systems.

Applying RL to skyline query processing presents a promising new algorithm to optimizing skyline query performance in large scale, uncertain or dynamic datasets. To handle the limitations of existing skyline queries, reinforcement learning offers a way to do this and leads to a more adaptive, strategic way to process skyline queries. In the RL algorithm, the agent can be created to sample the dataset and learn what kind of prioritization to apply to skyline points, through interacting with the data. In lieu of exhaustive comparison of all data points, the RL agent will be able to learn where in the dataset skyline points tend to be located more frequently and thus spend effort there rather than combatively comparing far too many data points to find out.

A RL algorithm that we can consider for skyline queries is the Policy Gradient algorithm, which learns a policy that maps states to actions directly (Zhang et al., 2020). In this case, the agent learns to find actions that increase the likelihood of identifying non-dominated points according to the experience of the dataset. The policy is updated weekly as the agent learns how to more effectively identify skyline points in complex high-dimensional datasets. In uncertain environments, the attributes of data points may not be fixed, but rather probabilistic, and skyline queries are particularly effective when calibrated with RL. For instance, in a transportation network, in which travel times and costs can change with time due to traffic or weather conditions, ranking then can be unreliable due to an inability of existing skyline algorithms to compensate for such a system's inherent uncertainty. However, RL agents are adaptable to such they can learn to deal with such uncertainty over time. The agent also learns in which ways data point variability affects their dominance relationships and which data points are likely to be part of the skyline. This makes the outcome of the RL-based skyline query process more robust and more accurate when the data is uncertain, unreliable, or varies over time.

Given its advantages, we propose to integrate reinforcement learning (RL) into Graph Neural Networks (GNN) to improve skyline query processing in uncertain and large-scale graph databases. GNN and RL each have strengths that work together with one another, and has the potential to increase the efficiency, scalability, and accuracy of skyline

query demand in complex scenes (Munikoti et al., 2023). GNN performs better than almost every computational algorithm in terms of node classification, link prediction, and graph traversal due to the particularity of the structural information included in the graph. However, RL suits well for dynamic decision-making in which an agent infers from interactions with an environment what its actions should be over time. Using GNN along with RL capability, it was possible to boost the agent's capability of learning from both the graph structure as well as the rewards from the environment to make more reasoned decisions when querying the skyline.

RL agents can learn to prefer a certain region on the graph by the learnable node embeddings, and consequently, they can concentrate on regions where the skyline points are potentially more likely to reside. Therefore, it reduces the amount of effort required for comparing entire works, and consequently increases the efficiency of running a query. An additional advantage of combining GNN with RL is that uncertain data may be processed. GNN can learn to represent the uncertainty in the data when the attributes in graph databases are stochastic and express it with probabilistic distributions. With this information, the RL agent can then make decisions considering uncertainty, learning to avoid regions of the graph where the uncertainty is high and in regions where the probability of finding skyline points tends to be higher. Thus, the combined GNN-RL model performs well for probabilistic skyline queries, which seek nodes that are more likely to be in the skyline, given their probabilistic attributes.

2.2.2 Evaluation Metrics for Skyline Queries

2.2.2.1 Performance Evaluation Metrics

With large-scale graph databases, evaluating the proposed skyline query model requires multiple performance metrics so that researchers can get a complete picture of its

effectiveness. Metrics like accuracy, precision, recall, F1 score, and ROC-AUC are indicative of a spectrum of dimensions through which researchers will evaluate how good the proposed model is. These metrics are usually applied to classification problems but can be adapted to skyline queries by treating the point identification problem itself as a binary classification issue.

The most common metric is “accuracy”, the ratio between correctly identified skyline points and correctly identified non-skyline points to the total number of points (Vujovic, 2021).

$$\text{Accuracy} = \frac{True\ Positives + True\ Negatives}{Total\ points} \tag{1}$$

Accurate results are computed as a number of right predicted points divided by the number of points in the total dataset. Such a measure gives quick feedback about how well the model works. Unfortunately, in the case of imbalanced datasets, accuracy can be misleading. For instance, in a dataset where we only have 5 percent of points are skyline, a model that predicted all points as non-skyline can achieve 95 percent accuracy, but it will not perform well in filtering out any skyline points. Generally, accuracy is used to start with and measure any classification model including skyline query one. This gives us a view of the correctness of models. In skyline query scenarios, however, which are usually very resource-constrained, accuracy alone is not sufficient. To complement Accuracy, “Precision” also needed to be calculated in terms of skyline points.

$$\text{Precision} = \frac{True\ Positives}{True\ Positives + False\ Positive} \tag{2}$$

It represents a ratio of true positives to the sum of true positives with false positives. The identified skyline points with high precision, and are likely to be relevant, reducing the likelihood of non-skyline points clattering the results. It is particularly important when making decisions (Yacouby & Axman, 2020). When the cost of false positives is high, precision is important. Introducing irrelevant points to skyline results by false positives increases the complexity of either decision making process. For this reason, precision in skyline queries is crucial to produce only relevant points.

Sensitivity, or recall, is the proportion of actual skyline points correctly identified by the model. We calculate recall to be the number of true positives divided by the number of false negatives and true positives.

$$\text{Recall} = \frac{True\ Positives}{True\ Positives + False\ Negatives} \tag{3}$$

The high recall indicates that the model is able to capture about majority of relevant skyline points. However, when missing a potential skyline point could lead to bad decisions, a high recall value is crucial (Powers, 2020). As is the case when false negatives incur a high cost, recall is also important. Not accounting for a skyline point can result in indices being taken that aren't optimal. In this case, high recall guarantees to find all relevant skylines, even if this means including a few false positives

As a balance metric between precision and recall, the F1 score is the harmonic mean of the two (Mortaz, 2020). Though it is only one metric, it is useful in skewed datasets where we only have to balance precision vs recall.

$$\text{F1 Score} = 2 \times \frac{Precision + Recall}{Precision \times Recall} \tag{4}$$

The higher the F1 score, the better the performance of the model in finding skyline points. By being the harmonic mean of precision and recall the F1 score offers a balanced view of model performance. It is very useful in imbalanced datasets where it is very important to ensure low false positives and false negatives.

The ROC-AUC score indicates the model's ability in applying various threshold values and distinguish skyline points from non-skyline points. The ROC curve is drawn between the true positive rate and the false positive rate, with the AUC the area under the curve (Carrington et al., 2022). A higher AUC implies that the model tends to better discriminate between classes. Robust metrics validation such as K-Fold cross-validation is also used to get reliable estimates of model performance under a range of data distributions by dividing the dataset into K subsets and training the model on K-1 subsets and testing it using the remaining fold. Researchers have used 10-fold for the proposed model performance validation (Zhang & Liu, 2023).

Hypervolume (HV) captures the volume in the objective space that is dominated by the predicted skyline set relative to a reference point (Guerreiro et al., 2021). It simultaneously reflects the breadth of coverage and the quality of the solutions in the skyline. A higher hypervolume indicates that the predicted skyline covers a large and relevant portion of the true trade-off surface, which is desirable for providing diverse and optimal choices.

Generational Distance (GD) measures the average closeness of the predicted skyline points to the true Pareto front (Retzlaff et al., 2024). Low generational distance signifies that the model’s predictions are near-optimal in the attribute space, thus closely approximating the best trade-offs possible. This metric ensures the accuracy of the skyline points, beyond simple classification correctness.

Spacing (Sp) assesses the uniformity of distribution of predicted skyline points along the Pareto front (Zheng et al., 2016). Uniform spacing means the skyline points are well spread out, preventing clustering that could bias decision-makers towards certain regions of the solution space. Good spacing ensures stakeholders receive a representative set of options covering the entire spectrum of trade-offs.

Precision@10 measures the proportion of true skyline points within the model's top 10 predictions. It reflects the model's effectiveness at correctly identifying the most important points when only a small subset of the skyline is considered, which is often the case in real-world decision-making due to cognitive or resource limitations.

Recall@10 This quantifies how many true skyline points are captured within the top 10 predictions relative to all true skyline points. Although typically low due to the limited shortlist, high recall@10 indicates that the model successfully includes a meaningful portion of key skyline points even in a highly constrained selection.

Average Precision (AP) summarizes the precision values achieved at varying levels of recall (Carrington et al., 2022). It provides an integrated assessment of the model's ranking performance over the entire prediction range. This is particularly important in imbalanced skyline datasets, where maintaining high precision across different recall thresholds is challenging.

Area Under the Precision-Recall Curve (AUPRC) offers a single-value measure that captures the trade-off between precision and recall across all classification thresholds (Beam, 2025). It is especially valuable in evaluating performance in datasets with skewed class distributions like skyline queries, where standard ROC metrics may be less informative.

Normalized Discounted Cumulative Gain at 10 (nDCG@10) evaluates not only the presence of true skyline points in the top 10 but also rewards correct ranking order, placing more important points higher in the list (Wang, Wang, et al., 2013). This metric aligns closely with user experience in decision-support systems, where the rank order can influence choice effectiveness. F1-Score Drop (F1-drop) captures the decrease in F1-score when Gaussian noise is added to node features and when a fraction of edges is randomly removed to simulate structural corruption (Jiang & Tsai, 2025). Measuring the F1-drop quantifies the resilience of the algorithm's latent representations and decision-making ability under uncertainty and perturbations, validating its suitability for real-world dynamic environments.

As a whole, these performance metrics together represent a general algorithm to evaluate skyline query models of large-scale graph databases and allows for a detailed research of the performance of skyline query algorithm.

2.3 Classification of Algorithm Integration

Some complex problems in computer science and artificial intelligence call for the combination of multiple algorithms to arrive at solutions. With progress in research, the demand for improving computational models results in the setup of algorithms which combine multiple algorithms utilizing the benefits and lessening their deficits. Two such type of algorithm can be stated as combination algorithms or hybrid algorithms. Although the two processes include combining various methods together, the manner in which the integration happens and the amount of interdependence among parts of components varies. The combination of algorithms, usually, occurs when multiple algorithms are connected in sequence or in parallel without altering the internal detail of the algorithms while a Hybrid Framework directly combines other techniques in such way that one algorithm affects the other internally.

2.3.1 Combination Algorithm

Combination algorithm is defined as combining more than one independent algorithm with little or no interaction. The various algorithms continue to have their own individual functions, and usually it is their function that is applied in sequence or in parallel. The main reason to use combination algorithms is to gain the benefits of multiple techniques while not fundamentally changing how they work inside.

An example of combination algorithm is one of the methods of ensemble learning in deep learning (e.g., bagging, boosting). In these methods, one trains independently multiple models and aggregates their predictions to improve the overall accuracy. Another example is to combine heuristic and exact algorithms in solving optimization problems: first solving the problem using a heuristic method, which returns an approximate solution, and then using an exact method to improve the solution. For such cases, the combination takes each method's strength and does not alter their fundamental operation in any way. A combination algorithm is distinguished primarily by having independent parts that generate outputs that are either combined into one output or used in sequence to give an overall better output. Although this approach works very well in many cases, for deeper integration it's not suitable, as it can be used to generate more adaptive and intelligent models.

2.3.2 Hybrid Framework

A Hybrid Framework is a more elaborate process that combines different algorithms by one method continually affecting the procedure of the other method. Unlike combination algorithms, where the individual components operate independently, Hybrid Frameworks are synthesized to dual interface various methodologies such that they virtually 'completed' and 'enhanced' each other in a dynamic manner.

One of the examples of Hybrid Framework is training neural networks by reinforcement learning. In such a system, reinforcement learning (RL) not only is combined with a neural network but they actually affect the learning itself by adjusting the model parameters in terms of rewards. Since this makes the two components interdependent it becomes a mechanism for adaptive learning. Another example is the combination of evolutionary algorithms and deep learning in which evolutionary strategies act to tune the hyperparameters of the network in the process of learning which is faster. The most distinguishing feature of Hybrid Frameworks is that they inherently build smarter, context aware solution. Hybrid approaches combine two or more approaches through the embedding of one technique inside another in order to provide systems with the ability to learn and adapt in ways that a simple sum of methods cannot achieve. They are, however, more complex to design and care must be taken when tuning to guarantee stability and efficiency.

2.4 Research Gaps and Opportunities

The reviewed literature reveals a clear disconnect between algorithmic advancements in skyline query processing and the practical demands of large-scale uncertain graph databases. In particular, existing methods lack integrated learning-based mechanisms that jointly address uncertainty modeling, scalability, and adaptive optimization.

2.4.1 Major Challenges

Data uncertainty handling is one of the main obstacles when it comes to implementing skyline in large and uncertain graphs. In most real-world graphs, the attributes are deterministic (Salamanos et al., 2017). Consider a transportation network, where for instance travel times vary from very high to very low due to traffic conditions or road closures. The evaluation of dominance relationships is made difficult by this uncertainty

and thus algorithms to calculate skyline queries in the presence of probabilistic attributes are needed (Khalefa et al., 2010). These probabilistic comparisons are difficult for existing algorithms as comparisons involve dominance between probability distributions rather than specific values.

Challenges are further exacerbated by the size and complexity of large-scale graphs. In modern graph databases, we often have millions or even billions of nodes and edges and efficient algorithms for traversing this kind of graph and deciding dominance relationships are needed (Walke et al., 2023). Skyline queries computation expenses raise a lot if the dataset size is large, or when probabilistic attributes are incorporated (Lai et al., 2020). Another major difficulty in processing skyline query is that the number of candidate objects can be high. However, in many graph databases we typically have nodes and edges associated with multiple attributes, leading to a high dimensional space where the number of skyline points can grow exponentially. High dimensional constraints which is according to Keogh and Mueen (2017), referred to as the 'curse of dimensionality' can make the computation of skyline queries using existing algorithms not effective. Additionally, having many skyline points can overwhelm users and can make it hard to make decisions out of these outcomes.

On both sides, skyline query processing is made more complex by the distributed nature of modern graph databases. To improve scalability and performance, many large-scale graphs are stored across multiple machines or servers. Such distribution requires coordination between nodes because skyline points may be distributed across partitions of the graph. Exchange of information about partial skyline points is necessary which add considerable communication overheads that can make slow query processing (Li et al., 2020).

Lastly, real world applications present ever changing data and as new data is introduced and existing data is modified, the graph database structure and relationship also changes.

For example, skyline queries for a transportation network need to be updated in real time as new routes are added or travel conditions change. Updating these queries in dynamic environments is a difficult problem, especially in uncertain graphs where probabilistic dominance relationships need to be computed anew in response to each modification (Banerjee et al., 2020). Our challenges, therefore, occur for managing uncertainty in skyline queries in uncertain large scale graph databases, while considering data uncertainty, computational efficiency, high dimensionality, distributed architectures, as well as the dynamic nature of the data. This requires efficient and scalable algorithms and techniques for addressing uncertainty in complex environments

2.4.2 Summary of findings

Table 2.1 A short summary of findings from literature

Paper	Uncertainty	Dataset Used	Dataset Name	Method	Result
Probabilistic Skylines on Uncertain Data (Pei et al., 2007)	Yes	Real & Synthetic	NBA	Bottom-Up and Top-Down	The proposed algorithms are significantly faster than exhaustive methods and scalable for large datasets
A Performance Analysis of Prediction Techniques in Handling High-Dimensional Uncertain Data for the Application of Skyline Query Over Data Stream (Mohamud et al., 2024)	Yes	Real & Synthetic	NBA, Synthetic	Linear Regression, k-NN, Random Forest, Decision Tree, CRM	CRM method outperformed other techniques in terms of execution time, RMSE, precision, recall, and F1-score.
SUBSTITUTION: An Efficient Algorithm for Probability Skyline Queries on Discrete Uncertain Data (Ma et al., 2012)	Yes	Real	NBA	SUBSTITUTION Algorithm	SUBSTITUTION algorithm drastically improves the efficiency of skyline probability calculation, demonstrating superior performance compared to other methods.
Finding Probabilistic k-Skyline Sets on Uncertain Data (Liu et al., 2015)	Yes	Real & Synthetic	NBA, Synthetic	Heuristic pruning, Layered range tree, Baseline algorithm	Efficient algorithm for k-skyline sets, faster than baseline methods with scalability demonstrated on both real and synthetic datasets.
The Indistinguishability Query (Lall, 2024)	Yes	Real & Synthetic	NBA, Island, House	Squeeze-u, MinR, MinD, UH-Random	Squeeze-u achieves efficient ε-indistinguishability queries, significantly reducing error

					with fewer questions and demonstrating scalability for large datasets.
Finding Skylines for Incomplete Data (Bharuka & Kumar, 2013)	Yes	Real & Synthetic	MovieLens, NBA	Sort-based Incomplete Data Skyline (SIDS) Algorithm	SIDS algorithm efficiently computes skylines for incomplete datasets, outperforming ISkyline in scalability and memory utilization.
Preference Queries Processing over Imprecise Data (Khalefa, 2011)	Yes	Real & Synthetic	Synthetic and Simulated	ISkyline, UPref, PrefJoin, PrefJoin*	Efficient algorithms for skyline and preference queries over uncertain and incomplete data, improving scalability and reducing computation time.
Directional Queries: Making Top-k Queries More Effective in Discovering Relevant Results (Ciaccia & Martinenghi, 2024)	Yes	Real & Synthetic	NBA, Synthetic datasets	Directional Queries (based on weighted mean and distance from preference line)	Directional queries enhance the quality of top-k queries by favoring balanced results, outperforming linear and non-linear methods with minimal computational overhead.
Skyline Queries Computation on Crowdsourced-Enabled Incomplete Database (Swidan et al., 2020)	Yes	Real & Synthetic	Simulated databases, synthetic datasets	Approximate Functional Dependency (AFD)-based Estimation, Crowdsourcing	Efficient handling of incomplete skyline queries with reduced cost and latency; scalable approach leveraging AFDs and crowdsourced databases.
A Framework for Ranking and KNN Queries in a Probabilistic Skyline Model (Li et al., 2015)	Yes	Real & Synthetic	NBA, Synthetic datasets	Bounding-Pruning-Refining Strategy, Space Partition Tree (SPTree)	Efficient framework for ranking and KNN queries, significantly reducing computational costs compared to baseline methods, with demonstrated scalability.

Probabilistic Skylines on Uncertain Data: Model and Bounding-Pruning-Refining Methods (Jiang et al., 2012)	Yes	Real & Synthetic	NBA, Synthetic datasets	Bottom-Up, Top-Down, Hybrid Algorithm	Efficiently computes probabilistic skylines, significantly improving computational efficiency and scalability compared to exhaustive methods.
Skyline Query Processing for Clustering the Multidimensional Data (David & Jayachandran, 2016)	Yes	Real	MovieLens, Book Crossing	Skyline Query Processing, Adaptive Clustering	Solves the long-tail problem by clustering unpopular or new items using skyline queries, improving scalability and recommendation performance.
The Survey on Skyline Query Processing for Data-Specific Applications (Gothwal et al., 2018)	Yes	Real & Synthetic	MovieLens, NBA, Synthetic datasets	Various skyline algorithms, including Block-Nested Loop (BNL), Sort-Filter-Skyline (SFS), and Divide and Conquer	Provides a comprehensive survey of skyline query algorithms for diverse data types (uncertain, incomplete, time series), highlighting their efficiency and limitations.
Modeling and Computing Probabilistic Skyline on Incomplete Data (Zhang et al., 2019)	Yes	Real & Synthetic	MovieLens, NBA	SPISkyline, SPCSkyline, SPASkyline Algorithms	Proposed probabilistic skyline computation on incomplete data achieves high accuracy and efficiency, outperforming naive methods by tens of times.
Systematic Study of TKD Queries on Data, Which Involves the Data Having Some Missing Dimensional Values (Lakshmi & Rao, 2017)	Yes	Real & Synthetic	MovieLens, NBA, Zillow	ESB, UBB, IBIG Algorithms	Proposed algorithms efficiently address TKD queries on incomplete data using novel pruning techniques and electronic image compression for improved efficiency.
CIDS: An Efficient Algorithm for Processing Skyline Queries for Partially Complete Data in Cloud Environment (Gulzar & Alwan, 2022)	Yes	Real & Synthetic	NBA, MovieLens, CoIL 2000, Synthetic	Cloud-based Incomplete Data Skyline (CIDS) Algorithm	Efficiently identifies global skylines with reduced data transfer, processing time, and domination tests,

					outperforming existing methods.
Missing Values Estimation for Skylines in Incomplete Database (Alwan et al., 2018)	Yes	Real & Synthetic	NBA, CoIL 2000, MovieLens	Approximate Functional Dependencies (AFDs), Probability Correlations	Proposed approach estimates missing values in skylines effectively, achieving high accuracy and scalability on incomplete datasets.
A Synthesis Model for Multimedia Video Files in Different Views (Priya Dharshini & Velladurai, 2017)	No	Real	Multimedia datasets	Dynamic Programming Algorithm for Virtual View Synthesis	Proposed model effectively synthesizes virtual views for 3D scenes, improving navigation quality and reducing distortion under bandwidth constraints.
An Efficient and Scalable Location-Aware Recommender System (Srilakshmi & Sunil Kumar, 2017)	Yes	Real	MovieLens, Foursquare	LARS: User Partitioning and Travel Penalty Techniques	LARS system provides high-quality, scalable location-based recommendations, achieving twice the accuracy of existing systems for large datasets.
Understanding the Meaning of a Shifted Sky: A General Framework on Extending Skyline Query (Zhang et al., 2024)	Yes	Real & Synthetic	NBA, Synthetic datasets	Generalized Framework for Dominance Relationships, Cone Dominance, Mapping-Based Dominance	Proposes a flexible dominance framework to handle skyline queries with varying dominance relationships, achieving controllable size and robust results.
Generic Analysis and Methods for Computing Skyline Variants (Zhang et al., 2024)	Yes	Real & Synthetic	NBA, MovieLens, Synthetic datasets	Generalized Dominance Framework, Cone Dominance, Mapping Dominance	Proposed framework simplifies skyline variant computation, maintaining efficiency and adaptability across datasets, with extensive experimental validation.
An Energy-Efficient Skyline Query for Massively	Yes	Real & Synthetic	Simulated datasets	E2Sky (Node Cut and Tuple Cut Strategies)	E2Sky reduces transmission and computational costs, improving efficiency and

Multidimensional Sensing Data (Wang et al., 2016)					scalability for large-scale multidimensional sensing data.
Geometry-Based Distributed Spatial Skyline Queries in Wireless Sensor Networks (Wang et al., 2016)	Yes	Real & Synthetic	Simulated WSN Data	GDSSky: Convex Hull, Voronoi Diagram, Distributed Queries	GDSSky method efficiently computes skyline queries in WSNs, reducing energy consumption and improving scalability and accuracy for spatial skyline data.
Enhanced Distributed Dynamic Skyline Query for Wireless Sensor Networks (Ahmed et al., 2016)	Yes	Simulated	Simulated WSN Data	EDDS: Threshold-Based Hierarchical Approach in DBDCS	EDDS efficiently computes dynamic skylines in WSN, reducing transmission cost, energy consumption, and latency, while improving accuracy and scalability.
Continuous Probabilistic Skyline Queries for Uncertain Moving Objects in Road Network (Pan et al., 2014)	Yes	Real	Oldenburg Road Network, Cixi City Road Network	PSUR Algorithm, Trigger Events, Pruning Strategies	PSUR efficiently updates probabilistic skylines incrementally, outperforming baseline and priority methods in runtime and scalability.
Adaptive Processing for Distributed Skyline Queries over Uncertain Data (Zhou et al., 2016)	Yes	Real & Synthetic	Household (Hous), Synthetic datasets	IDSUD Framework, ADSUD Algorithm, Improved PR-tree	ADSUD reduces query time, I/O costs, and improves progressiveness and bandwidth efficiency, outperforming the e-DSUD algorithm in distributed environments.
Distributed Spatial Skyline Query Processing in Wireless Sensor Networks (Yoon & Shahabi, 2011)	Yes	Simulated	Wireless Sensor Network Data	Distributed Spatial Skyline (DSS) Algorithm	DSS algorithm reduces communication overhead by up to 91% over centralized approaches, achieving 100% accurate and progressive results for spatial skylines.

A Systematic Literature Review of Skyline Query Processing Over Data Stream (Mohamud et al., 2023)	Yes	Real & Synthetic	NBA, MOVIE, Synthetic Datasets	Sliding Window Approaches, Probabilistic Skyline Algorithms, Parallel Query Processing	Comprehensive review highlights advances, challenges, and future directions for skyline queries over data streams, emphasizing scalability and query efficiency.
Efficient Pr-Skyline Query Processing and Optimization in Wireless Sensor Networks (Li & Xiong, 2010)	Yes	Simulated	Wireless Sensor Network Data	SKY-SEARCH Algorithm, Distributed Optimization Strategies	SKY-SEARCH achieves efficient Pr-Skyline computation, significantly reducing energy, transmission, and storage costs while maintaining scalability.
Efficient and Progressive Algorithms for Distributed Skyline Queries over Uncertain Data (Ding & Jin, 2012)	Yes	Real & Synthetic	NYSE, Independent, Anticorrelated	DSUD Algorithm, e-DSUD Algorithm, Feedback Mechanism	e-DSUD reduces communication and computation costs significantly compared to DSUD, achieving progressiveness and scalability for skyline queries.
On Efficient Processing of Continuous Reverse Skyline Queries in Wireless Sensor Networks (Yin et al., 2017)	Yes	Simulated	Wireless Sensor Network Data	ECRS Algorithm, Mapping Scheme, Node Pruning Techniques	ECRS efficiently computes reverse skyline queries, significantly reducing communication cost and energy consumption while ensuring correctness.
Progressive Skyline Query Processing in Wireless Sensor Networks (Chen & Liang, 2009)	Yes	Real & Synthetic	Intel Lab, Synthetic datasets	a-FDP, g-FDP, α-DDP Algorithms	Proposed algorithms improve energy efficiency and prolong network lifetime while ensuring scalable and progressive skyline query processing.
M-Skyline: Taking Sunk Cost and Alternative Recommendation in	Yes	Real & Synthetic	CarDB, HotDB	IGP, EGP Algorithms, Complete Group Pruning	Efficiently handles skyline queries with sunk cost and alternative recommendations,

Consideration for Skyline Query on Uncertain Data (Zeng et al., 2019)					achieving scalability and significant reduction in computation time.
Optimizing Skyline Queries over Incomplete Data (Lee et al., 2016)	Yes	Real & Synthetic	MovieLens, NBA	SOBA Algorithm, Bucket-Level Optimization, Point-Level Optimization	SOBA reduces dominance tests by two orders of magnitude, improving scalability and effectiveness in retrieving meaningful skylines for incomplete data.
Reconciling Skyline and Ranking Queries (Ciaccia & Martinenghi, 2017)	Yes	Real & Synthetic	NBA, UNI, ANT, HOU	Restricted Skyline Operators (nd, po), F-dominance, Linear Programming	Introduces restricted skyline operators combining skyline and ranking queries, enhancing flexibility and scalability, with efficient query processing and pruning.
Probabilistic Skyline Queries (Böhm et al., 2009)	Yes	Real & Synthetic	NBA, Synthetic datasets	Priority Algorithm, Indexed Algorithm, Monte Carlo Integration	Efficient computation of probabilistic skylines, demonstrating scalability and significant reduction in runtime compared to baseline methods.
Ranking Uncertain Sky: The Probabilistic Top-k Skyline Operator (Zhang et al., 2011)	Yes	Real & Synthetic	NBA, Synthetic datasets	Exact Algorithm for Discrete Case, Randomized Algorithm with E-approximation	Efficiently computes top-k skyline objects for discrete and continuous uncertain datasets, significantly improving scalability and accuracy over existing methods.
TANGENT: A Novel, "Surprise-Me", Recommendation Algorithm (Onuma et al., 2009)	Yes	Real & Synthetic	MovieLens, CIKM, Synthetic Graphs	TANGENT Algorithm, Bridging Score, Random Walk with Restart	TANGENT broadens recommendation diversity, improving user satisfaction by combining relevance and connectivity, outperforming conventional methods.

A Survey of Queries over Uncertain Data (Wang et al., 2013)	Yes	Real & Synthetic	Sensor, RFID, LBS, Synthetic datasets	Comprehensive Analysis of Skyline, Top-k, NN, Aggregate Queries	Highlights advances, challenges, and future directions in uncertain query processing, focusing on scalability, diversity, and novel data management techniques.

2.5 Summary

The area of skyline queries in the context of uncertain and large-scale graph databases is rather promising and can be further developed, pointing to several directions that seem to be most important for overcoming the current challenges. A main concern is the efficient management of the uncertainty in skyline queries. The real-world graph databases consist of attributes and relationships that may have probabilistic or fuzzy nature and, hence, the need for developing sound algorithms that can analyze dominance relationships under such circumstances. Probabilistic algorithms, Bayesian systems, and deep learning algorithms appear to be viable solutions to this problem. One particular area that requires further exploration is the issue of performance optimization of skyline query algorithms for large graph datasets. This is a problem since graphs with millions or even billions of nodes and edges are not uncommon these days for graph databases. Likewise, the high-dimensional datasets are more complicated in terms of computation issues because the number of skyline points increases exponentially with an increase in the number of attributes.

Dynamic graph databases also present their own set of issues that need new algorithms to solve. The real-world graphs are dynamic, as new data are continuously added or updated, the skyline query algorithms must be modified to handle the real-time update efficiently. Algorithms and techniques of incremental processing and adaptive algorithms for dynamic skyline queries are crucial in providing efficient solutions. Moreover, deep learning can be meaningfully incorporated into the skyline query processing. Reinforcement learning, neural networks and their hybrid can be used to design intelligent and effective solutions that respond to the challenges of uncertain and dynamic graph-based environments. By leveraging advanced computational techniques, particularly from deep learning and distributed systems, the field can evolve to meet the demands of increasingly complex real-world applications.

CHAPTER THREE

RESEARCH METHODOLOGY

3.1 Introduction

In this chapter, the researchers present a methodology to evaluate deep learning algorithms developed for skyline query processing in large-scale uncertain graph databases. It intends to explain the process of implementation, testing, and comparison of the baseline and proposed framework. Three baseline algorithms are analyzed: these are the Top-K Skyline Objects Algorithm, meant to find the top-K skyline objects in databases with uncertain preferences, the ProbSky algorithm which treats the skyline query problem across distributed data environment in a probabilistic way, and the U-Skyline algorithm which is a novel framework of answering skyline queries in uncertain databases by taking uncertainty in attributes and relationships into account. Along with these pre-established algorithms researchers also assess how modern algorithms, in particular a hybrid framework that combines GNN and RL perform. GNN has the potential to model complex data point and data dependencies in ways pre-established algorithms cannot. Furthermore, the integration of reinforcement learning improves this process by enabling the framework to search through dynamic decision-making strategies that can help the framework navigate and process uncertain environments.

The core objective of the experiments carried out in this chapter is to assess the performance of these algorithms across key metrics that are critical for skyline query processing: The prediction model is based on accuracy, precision, recall, F1 score, and ROC-AUC. Baseline skyline algorithms such as TopK, ProbSky, and USkyline are tested in deterministic environments but their ability to cope with uncertainty and scale properly in large datasets remains unknown. In this chapter, researchers seek to highlight these

challenges and identify gaps that arise when baseline algorithms are applied where attributes of nodes and edges may vary and relationships may be volatile.

However, the experiments will simultaneously show how deep learning algorithms, more specifically GNN+RL hybrid framework, can tackle such challenges. The suggested alternative to the baseline processing algorithms of skyline queries is GNN which allows a better capability of learning and representing the underlying structure of the graph-structured data and consequently, more efficient and adaptable skyline query processing (Prakash, 2024). Finally, a reinforcement learning component further optimizes this framework with real-time generated improvements in the decision-making process based on real-time feedback from the environment. As a result, the experiments will examine whether deep learning-based algorithms can both accommodate uncertainty and deal with the complicated nature of large datasets better than baseline algorithms. One goal of the experiments is to offer a clear and comparative research of the baseline algorithms.

Researchers will conduct experiments on synthetic dataset to ensure evaluation over a broad range of conditions. Controlled experimentation made possible by synthetic datasets permits stress tests by introducing specific challenges (high dimensionality, severe uncertainty) to stress the limits of each algorithm.

In summary, this chapter establishes the methodology for a rigorous and comparative analysis of skyline query processing algorithms both baseline and deep learning based. Firstly, the aim is to demonstrate the limitations of baseline algorithms in uncertain and large-scale environments and explore the feasibility of proposed framework to solve these problems and help push forward research in the skyline query area.

3.2 Research Methodology and framework

This research adopts the Design Science Research (DSR) methodology to guide the systematic development and evaluation of a solution-oriented research artifact (Nitsche et al., 2021). The methodology is structured around three interrelated components: the Environment, the Research (Build–Evaluate cycle), and the Knowledge Base, connected through the relevance and rigor cycles.

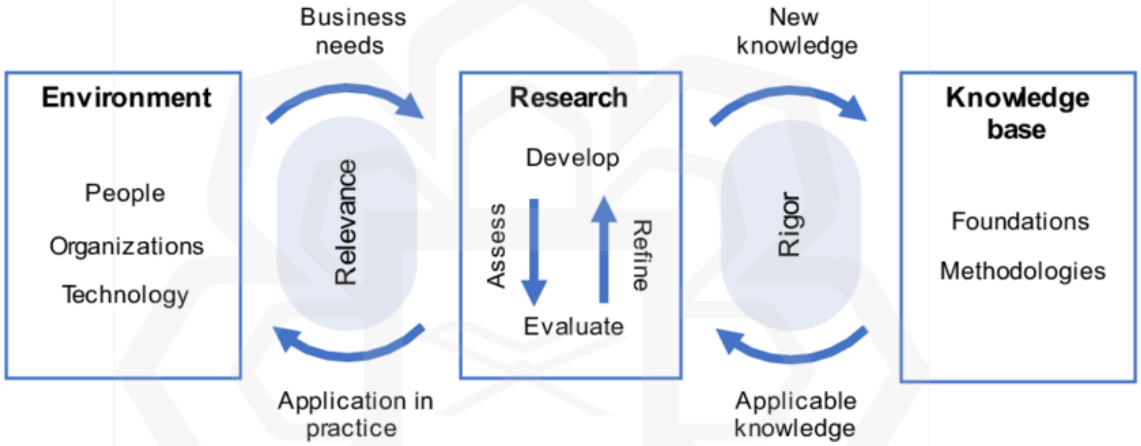


Figure 3.1 The design science research framework (Andersen, 2022)

The environment represents the practical problem context that motivates research, encompassing real-world needs and constraints derived from existing technological and organizational settings (Tuunanen et al., 2024). These needs inform the research objectives and define the requirements of the proposed solution. The research component constitutes the core iterative cycle in which artifacts are developed, evaluated, and refined through rigorous experimental and analytical procedures to ensure both effectiveness and validity (Kroop, 2025). Finally, the knowledge base provides theoretical foundations, established methodologies, and prior research that ground the study in scientific rigor, while also serving as the repository to which new insights and validated contributions are added (De Sordi, 2021). By integrating relevance-driven problem identification with rigorously

grounded design and evaluation, the DSR methodology ensures that the proposed research outcomes are both practically applicable and theoretically sound.

To clarify the logical structure of the research, the following conceptual framework summarizes the key inputs and assumptions, the proposed deep learning based processing mechanism, and the intended outputs and outcomes. In particular, it frames the problem as identifying relevant non-dominated solutions under uncertainty and motivates a learning-driven approach that can capture graph structure while adapting selection strategies to varying uncertainty conditions.

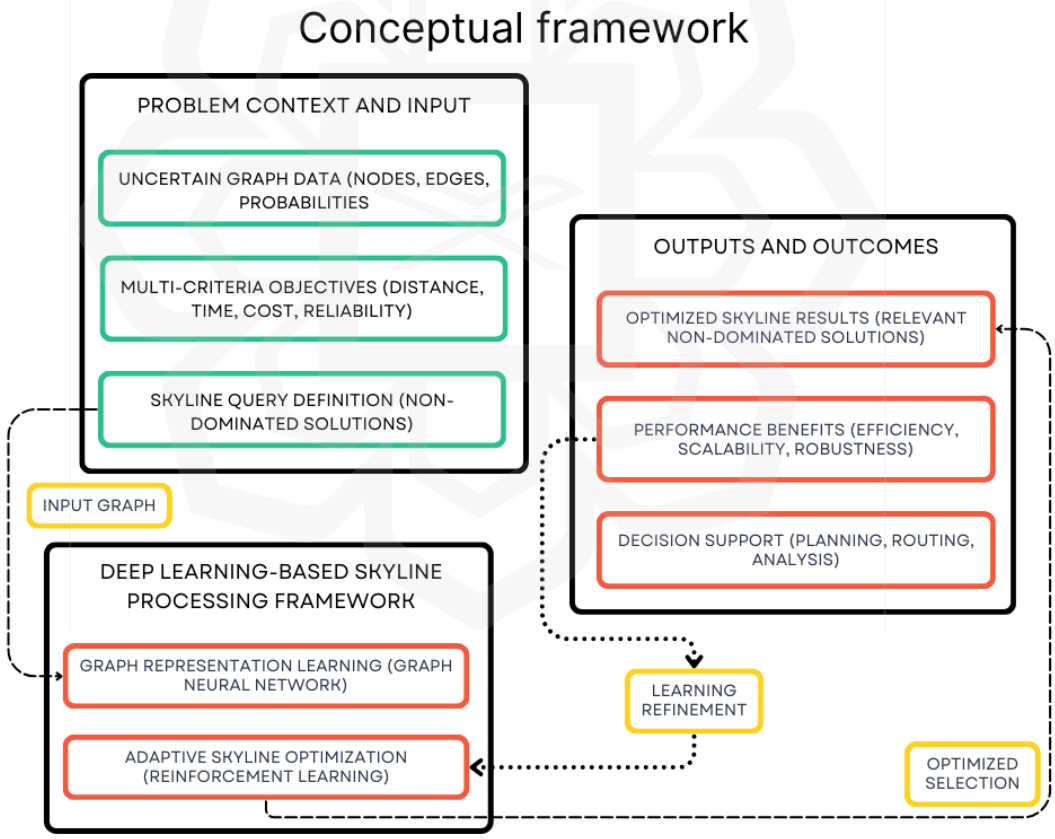


Figure 3.2 GNN + RL conceptual framework diagram

As illustrated in the conceptual framework, the input graph and skyline definition are processed through a two-part learning pipeline comprising graph representation learning and adaptive optimization. A Graph Neural Network (GNN) is used to learn embeddings that encode both topological dependencies and attribute patterns, while a Reinforcement Learning (RL) component guides the selection process toward high-quality skyline candidates under uncertainty. The refinement loop reflects iterative improvement of the learned representations and selection policy, leading to optimized skyline results and broader performance outcomes such as efficiency, scalability, and robustness. Collectively, these outputs enable decision support for downstream tasks by providing reliable non-dominated solutions and improved computational effectiveness in large-scale uncertain graphs.

3.3 Research Design

In the research design, the proposed methodology which solves research challenges in skyline queries can be considered systematic and comprehensive. Researchers begin their methodology with an extensive literature review to identify gaps and opportunities in baseline algorithms in dealing with large scale and uncertain graph data sets.

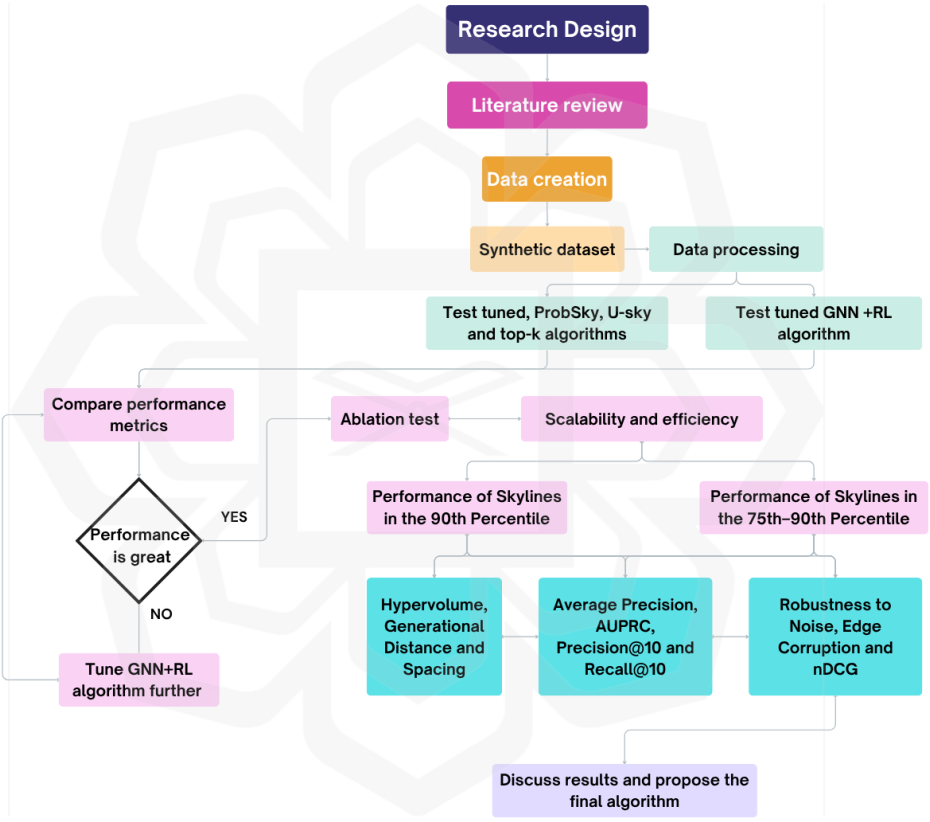


Figure 3.3 Research Design

The research methodology is structured around a systematic process that begins with the creation and preparation of synthetic datasets, followed by rigorous evaluation and iterative optimization of skyline query algorithms. Initially, synthetic datasets with well-defined characteristics are generated to serve as controlled testbeds for validating and

benchmarking various algorithms. These datasets undergo preprocessing and cleaning to ensure data quality and facilitate effective analysis.

Following dataset preparation, two parallel evaluation streams are executed. The first stream focuses on testing and tuning classical skyline query algorithms including ProbSky, U-Sky, and top K, assessing their performance on the synthetic datasets. The second stream concentrates on the implementation and evaluation of GNN+RL algorithm. This includes a hybrid GNN integrated with the Reinforcement Learning (RL) framework.

Comprehensive performance comparisons are conducted using an array of metrics such as accuracy, precision, recall, F1-score, and ROC-AUC. Based on these results, ablation studies are performed to investigate the impact of architectural components and training strategies on framework performance. Simultaneously, scalability and efficiency tests are carried out to examine the computational feasibility of the frameworks under varying graph sizes and densities.

The evaluation phase further explores the detailed performance of skyline predictions within critical percentile thresholds, particularly the 90th percentile and the broader 75th–90th percentile ranges. This involves calculating multi-objective quality measures such as hypervolume, generational distance, and spacing to quantify the accuracy and diversity of skyline approximations. Ranking and retrieval metrics, including Average Precision, AUPRC, Precision@10, Recall@10, and normalized Discounted Cumulative Gain (nDCG), are also assessed to gauge the effectiveness of candidate selections. Robustness evaluation is conducted to assess framework stability under noisy and corrupted data conditions.

Throughout this iterative process, performance comparisons guide the tuning of the GNN+RL framework to refine its accuracy and efficiency. The research culminates in a

comprehensive comparative analysis of all tested algorithms, highlighting their relative strengths and weaknesses. These insights inform the final proposal of a robust, efficient, and scalable skyline query solution for uncertain databases. This methodology embodies a cycle of testing, evaluation, and refinement, ensuring that the proposed GNN+RL framework is rigorously validated and optimized for real-world uncertain graph applications.

3.4 Critical Analysis and Novelty

The need to perform skyline queries to extract optimal solutions from multi-dimensional datasets makes them a critical tool for decision making. Unfortunately, as the data scale and complexity increases, especially in uncertain graph databases, baseline algorithms currently fall behind with respect to efficiency, accuracy, as well as scalability. In this section, researchers critically evaluate the limitations of selected skyline query algorithms and identify key research gaps and tries to highlight the novelty of proposed framework.

3.4.1 Critical Analysis of Existing Research

This thesis explores the use of Top-K Skyline, ProbSky, and U-Skyline algorithms in various domains, namely recommendation systems, transportation networks and environmental monitoring, where a lot of utility is shown in managing multi-dimensional datasets. Unfortunately, some inherent limitations constrain their usage in large and uncertain graph databases.

For instance, the Top-K Skyline algorithm was able to identify a subset of optimal points to a user defined criteria. But it does not scale high dimensionality data sets and large

graphs. As the dataset size grows, the computational cost of its approach raises dramatically rendering it impractical for real world large scale applications. The ProbSky is designed to answer probabilistic skyline queries by the use of enhanced pruning techniques. While this improves efficiency, it is limited in its ability to adapt dynamically to changes in uncertainty, which is a critical requirement in environments such as transportation networks where conditions frequently fluctuate. U-Skyline incorporates skyline methodology in a similar approach, whilst also extending skyline methods by allowing the inclusion of uncertain data attributes and relationships. While researchers structure a probabilistic model for skyline dominance, its effectiveness diminishes over extremely large datasets or graph structures which are highly interconnected.

Additionally, the majority of existing algorithms add significant dependence on pairwise comparisons to define the dominance relationship, thereby suffering from exponential growth in the computational complexity with the graph size and dimensionality. However, these algorithms are not built with mechanisms to dynamically learn and adapt to probabilistic characteristics of data in the uncertain graph databases, leading to their reduced accuracy and efficiency during uncertain graph database processing. For these reasons, these methods generally do not suffice for meeting the challenging requirements of typical modern applications like disaster management and intelligent transportation systems, which require robust and scalable solutions to handle uncertainty.

3.4.2 Novelty of the Proposed Framework

The proposed framework introduces a novel hybrid framework that combines the strengths of Graph Neural Networks (GNN) and Reinforcement Learning (RL) to address the limitations of existing methods. Unlike baseline algorithms, the GNN+RL framework is able to leverage the structural properties of graph databases to learn arbitrary complex patterns and relationships amongst nodes and edges. GNN aggregates information from the

neighboring nodes and propagates this knowledge in the graph, so the framework can capture both local and global dependencies which greatly improves accuracy of skyline point identification.

Reinforcement Learning fills the gap by dynamical tuning query execution based on the feedback from the environment. To reduce the amount of computation and scale up for a larger graph researchers show how the RL agent learns to prioritize the regions of the graph where more skyline points are likely to exist. Furthermore, the integrated framework also handles uncertainty explicitly while learning and deals with probabilistic data attributes and dependencies in a robust manner. The benefit of this is that when you're operating in a dynamic environment with conditions that change frequently, such as in a transportation network, the ability to perform decision making in real time, with real information is very valuable.

However, the proposed framework has several key innovations compared to baseline methods. First, instead of performing pairwise comparisons, researchers use GNN to predict dominance relationships to reduce computational cost. Second, through RL it dynamically adapts to changes in both the data attributes and the data relationships and is adaptive and resilient in uncertain environments. Lastly, with the help of the GNN's computational efficiency, and the strategic decision making of RL, the hybrid framework presents a scalable solution for large scale graph databases.

The novel GNN and RL integration for skyline query processing is a major advance. The proposed framework fills the gaps that exist in scalability, uncertainty handling, and dynamic adaptability, providing a robust and practical approach to handle complex real-world applications leading to more effective Multi criteria decision making in large scale uncertain graph databases.

3.5 Data Creation

3.5.1 Synthetic Data for Skyline Queries

3.5.1.1 Importance of Using Synthetic Data for Testing

The choice of data that is being used is crucial when evaluating the effectiveness of skyline query algorithms as they have to provide meaningful insights into their performance. The proposed framework is tested and validated using synthetic dataset for a comprehensive assessment.

Controlled experimentation is one of the main reasons for using synthetic data in testing. Unlike real-world data, which has all the associated challenges such as noise, missing values, and inherent uncertainty, synthetic data provides controlled environments for manipulation of attributes, and distributions to be precisely manipulated. Synthetic data lets researchers flexibly simulate scenarios by carefully designing the attribute distributions, attribute correlation, and size of the dataset (Suo et al., 2021). The controlled environment helps to isolate and test the performance of skyline algorithms, in particular their sensitivity to a variety of conditions, such as increasing dimensionality, reducing dominance levels, or even the addition of noise in the dataset. Synthetic data also enables researchers to perform stress testing of their framework, by synthesizing edge cases or extreme cases that do not naturally occur in reality (Jordon et al., 2022).

3.5.1.2 Synthetic Data Overview

For the evaluation of skyline query processing algorithms, the use of synthetic datasets plays a critical role in ensuring rigorous and systematic testing. While real-world datasets

capture complex, noisy, and practical scenarios, they often present challenges for controlled experimentation due to inherent variability in dimensionality, attribute distributions, and skyline point prevalence (Duan et al., 2011). Such uncontrolled factors can hinder the ability to isolate and analyze specific algorithmic behaviors.

In contrast, synthetic datasets offer the advantage of precise control over key experimental variables, including the number of dimensions, attribute correlations, and the density of skyline points. This level of control is essential for stress testing algorithms under diverse and extreme conditions, such as high dimensionality, strong attribute correlations, and varying sparsity or density of skyline points. Synthetic data therefore facilitates detailed scalability analyses, performance evaluation in controlled environments, and the identification of factors that directly influence skyline query processing efficiency.

Accordingly, this research exclusively employs a synthetic dataset designed to reflect realistic network and transportation conditions while affording flexibility in data generation. The dataset comprises multiple nodes and edges, enriched with attributes such as distance, probability, travel time, congestion levels, and weather impact. These features were selected both to emulate typical real-world factors influencing skyline points and to enable systematic variation in the data structure for comprehensive algorithm testing. By relying solely on synthetic data, this research ensures reproducibility and precise benchmarking of skyline query algorithms across a wide range of controlled scenarios.

3.5.1.3 Synthetic Dataset Generation Process

For this research, a synthetic dataset was generated specially for evaluating skyline query algorithms under control but realistic conditions. Researchers carefully designed the synthetic data generation process to produce a network of entities (nodes) connected by relationships (edges) where each entity has multiple attributes. This dataset uses a chosen

graph structure with 5,000 nodes and an arbitrarily specified edge density of 0.01, that is, approximately 0.01% of all possible node pairs are connected by edges. And so, we get a sparse graph, which is the most common type of graph seen in real-world scenarios, like transportation networks where not every single node is linked.

Table 3.1 Feature description table

Feature Name	Type	Description	Range
Source	Integer	Node ID of the starting point of an edge	0 to 4999
Target	Integer	Node ID of the destination point of an edge	0 to 4999
Distance	Integer	Distance between source and target nodes	1 to 100
Probability	Float	Probability of successfully traversing the edge	0.5 to 1.0
Travel Time	Integer	Estimated travel time for the edge	1 to 120
Congestion Level	Integer	Congestion level on the edge (traffic conditions)	1 to 10
Weather Impact	Integer	Impact of weather conditions on the edge	1 to 5
Label (Skyline)	Binary (0/1)	Indicates if a node is a skyline point (1 = Skyline)	0 (Non-Skyline), 1 (Skyline)

This synthetic dataset simulates a logistics network where a company needs to determine the best locations for placing distribution hubs in a city. Since products must be transferred between hubs multiple times a day, selecting hubs in strategic locations is critical. The skyline query results help identify the best hubs based on multiple road network factors, such as travel time, congestion, distance, and weather impact. This ensures that goods move efficiently across the network, reducing delays and improving logistics operations.

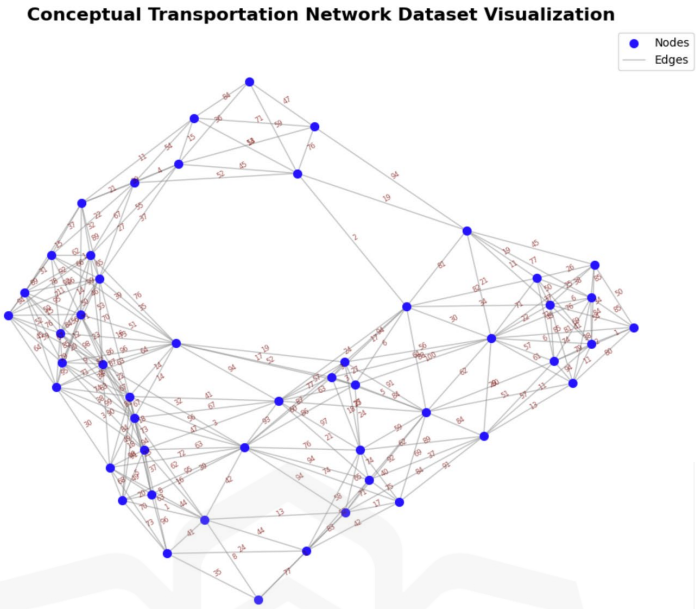


Figure 3.4 Conceptual transportation network dataset visualization

A random sparse adjacency matrix was created with each entry being an edge between two nodes for creating the dataset. Distance, traversal probability, travel time, congestion levels, and weather impact are the attributes set by the researchers that associate with these edges. The dataset was structured as follows:

- Nodes: 5000 nodes representing entity in a transportation network graph.
- Edges: An approximately 250,000 edge random sparse adjacency matrix was made by calculating the edge density at 0.01. There is a node for each object that has multiple attributes that are used in the evaluation of the skyline query.
- Attributes:
    - Distance: Values randomly generated from 1 to 100 representing the physical or temporal distance between any 2 nodes.
    - Probability: random values with a uniform distribution between 0.5 and 1. This attribute characterizes uncertainty in a relation like the possibility that an edge can be traversed or not.

- Travel Time: They are generated as random values between 1 and 120, correlated with distance. The greater the distance between nodes, the slower the travel time, but capped to be realistic.
- Congestion Level: Random integer values representing different levels of traffic congestion. The smaller the values the less congestion there is.
- Weather Impact: Simulating the impact of weather on edge usability by randomly mapping integer values between 1 and 5.

These attributes were picked based on their application to real-world problems especially domains of transportation. To inject reality into the data, the travel time attribute was made to have relation with distance, since travel times generally increase with longer distances, but may be subject to congestion or weather.

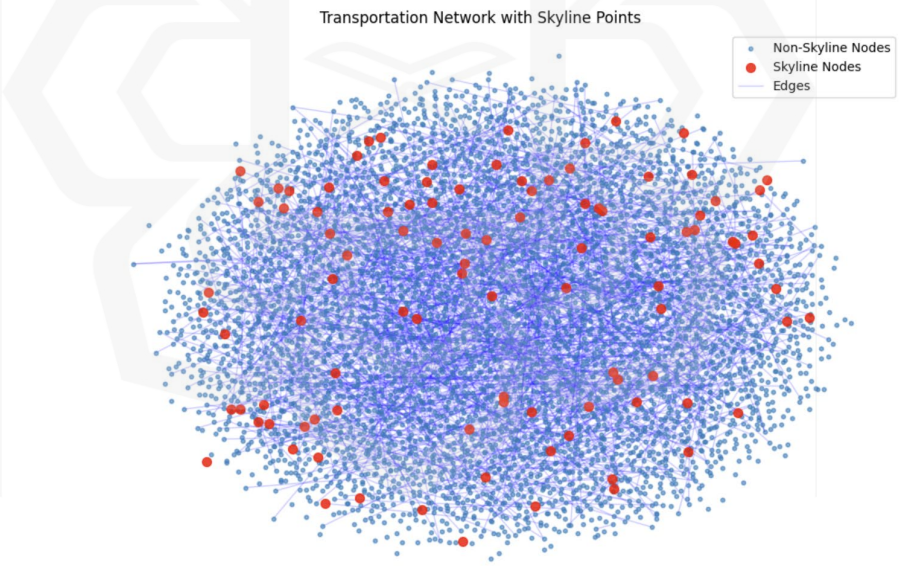


Figure 3.5 Synthetic transportation network with skyline points

Nodes were also labeled as skyline points and labels were also assigned to nodes. Of the 5,000 nodes, 100 were randomly selected to use as skyline points. To ensure that these nodes were great representations of the optimal ones in the dataset (and not just arbitrarily

good), researchers assigned more favorable attribute values to these nodes. For instance, skyline points had shorter distances, higher probabilities, lower travel times, lower congestion, and zero weather effects.

3.5.1.4 Data Preprocessing and Normalization

A few preprocessing steps were applied before inputting the data into the skyline query algorithms, including ensuring that the data was properly standardized, cleaned, and structured so researchers could begin their modeling. Any deep learning or skyline query pipeline starts with data preprocessing, which is important to make sure models get consistent and meaningful input. The following steps were undertaken:

1. **Normalization of Features:** Since each attribute of the dataset (e.g. distance, probability, travel time, congestion level, and weather impact) is on a different scale it is important to normalize the dataset such that any individual attribute does not dominate the model (Singh & Singh, 2019).
2. **Handling Missing Data:** In real-world datasets, missing values are always present. This was taken care of by the data generation process by modestly adding some random generation of attributes so that there weren't any missing values in the end.
3. **Oversampling of Skyline Points:** For large datasets, skyline points may represent a minority class, so oversampling was used on the dataset to balance it. To prevent the models from being biased towards non-skyline points, the majority class cardinality was adjusted by oversampling the minority class using the RandomOverSampler technique so the skyline examples would be sufficient to learn from (Frederickson & Polikar, 2018).
4. **Edge Index Creation:** For GNN+RL, researchers need a way to represent the graph structure explicitly through an edge index, denoting what connections there are between the nodes. The source target relationships contained in the adjacency

matrix of the edges were used to create the edge index, which correctly represents the structure of the graph when input to the GNN+RL algorithm.

To evaluate these skyline query algorithms on the dataset, these preprocessing steps were necessary. With normalization for the features, balancing the data in some way if not fully, and encoding the graph structure appropriately, researchers were then able to give the models high-quality input, resulting in more reliable and thoughtful results.

3.6 Artifact Design: Skyline Query Processing

3.6.1 Overview of Skyline Query Processing Algorithms

This section introduces the four main algorithms evaluated in this research for skyline query processing: A Graph Neural Network (GNN) combined with Reinforcement Learning (RL) hybrid framework, ProbSky Algorithm, U-Skyline Algorithm and lastly Top-K Skyline Objects Algorithm. Each algorithm is developed to solve the skyline query problem with varying algorithms on multi-criteria decision making, scalability, and uncertainty in large databases.

The Top-K Skyline Objects Algorithm is a deterministic Algorithm whose aim is to find the Top K skyline points ordered according to a pre-specified ordering of the attributes. They assume that users are usually interested in a few most related skyline points and not the entire skyline. This algorithm is intended for applications where the results must be concise, and scaling the number of skyline points becomes prohibitive in high-dimensional data (Sukhwani et al., 2021).

ProSky Algorithm takes a probabilistic algorithm to skyline queries, modeling uncertainty in the data points' attributes explicitly. It is more efficient for large-scale systems where the data is scattered across multiple nodes. This algorithm applies to situations in which attribute values are not fixed but rather follow certain probability distributions in which the situation under uncertainty is subject to real-time decision-making (Kuo et al., 2022). The U-Skyline Algorithm adapts the notion of skyline query to leverage uncertainty in both node attributes and relationships. The design is specifically for uncertain graph databases, where there is no determinism in the connections between data points (Liu et al., 2013).

The GNN + RL hybrid framework finally combines the Graph Neural Networks to learn the structural dependencies in graph-structured data and Reinforcement Learning to optimize the skyline query process. GNN also excels at tasks where it must reason about complex relationships between entities, and combined with RL, it can learn how to dynamically improve the decision-making process when feedback is provided in an uncertain environment. Researchers find that this algorithm is particularly useful for large-scale, uncertain graph databases where baseline algorithms struggle with scalability and uncertainty.

3.6.2 Baseline Skyline Algorithms

3.6.2.1 Top-K Skyline Algorithm Overview

The Top-K Skyline algorithm has been devised to determine the most important skyline objects from datasets which contain uncertainty. In an uncertain graph the nodes are usually data elements which contain probabilistic features, and the edges describe connections or impacts. In this section, the Top-K Skyline algorithm for uncertain graph systems is given in pseudocode. The algorithm targets at acquiring K skyline nodes that are based not only

on the attributes of the nodes but also on the graph topology. This helps in selecting the most important skyline nodes while at the same time taking data uncertainty into account. Sorting, pruning, and iterative refinement are used in the algorithm in order to optimize the computation process.

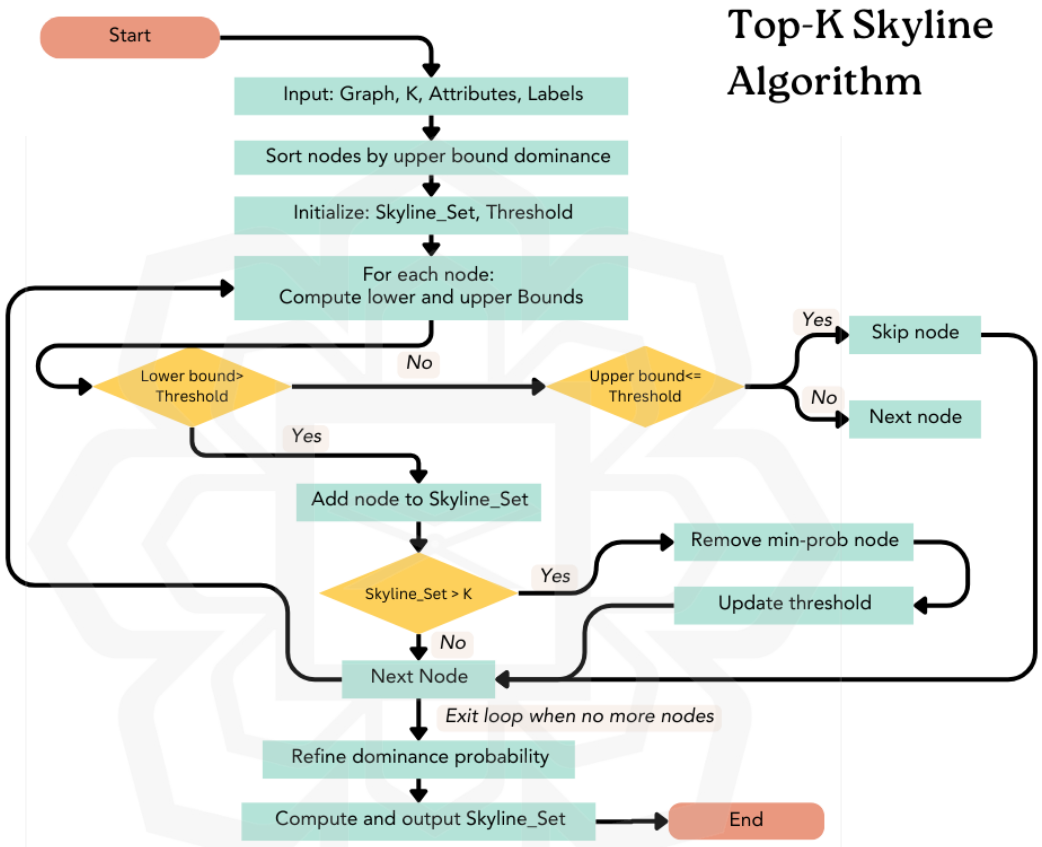


Figure 3.6 Flowchart of Top-K Skyline Algorithm (Sukhwani et al., 2021)

The algorithm first sorts the nodes in the graph in increasing order of their upper dominance bounds. The upper bound captures the probability that a node can be part of the skyline objects based on its attributes. Then, a candidate skyline set is set to be an empty set, and a threshold is employed to capture the minimum dominance probability for the candidate skyline element. The algorithm operates on nodes in a manner of declining upper bounds sequentially. For each node its lower and upper bounds of dominance probability

are calculated. When a node's lower bound is higher than the threshold, this node is considered as a candidate skyline node. If adding this node causes the set size to go over the value of K, then the node with the lowest probability will be removed to satisfy the Top-K constraint. Finally, based on the graph-based relationships and uncertain attributes of the candidate skyline set nodes, the dominance probabilities of the nodes are further computed. This enhancement guarantees that the skyline set that is generated at the end comprises the most relevant Top K nodes.

Top-K Skyline algorithm offers an efficient technique of determining the K most significant skyline nodes in uncertain graph environments. Its systematic way of sorting, pruning and refining makes it a very efficient in its computation while at the same dealing with the issues of uncertainty and big data. This algorithm is critical in decision support systems operating within dynamic and imprecise conditions (Sukhwani et al., 2021).

3.6.2.2 ProbSky Algorithm Overview

ProbSky is an effective and fast algorithm to compute probabilistic skyline queries in datasets with uncertainty. Here, the general idea of the ProbSky algorithm, its pseudocode adapted for an uncertain graph, and its usage for skyline query processing has been described by the researchers for a better insight later on. To minimize the computational cost, the ProbSky algorithm employs pruning that removes nodes which may not be skyline objects. The algorithm uses both local and global dominance relationships within the graph while exploiting probabilistic attributes in order to refine the skyline result set. The algorithm starts by assigning a small set of nodes which are based on the attribute bounds and dominance probabilities. Those nodes which are dominated by other nodes, or which have probabilities less than a given cut-off value are removed from the computation at an early stage. The rest candidate nodes are checked one by one to calculate their exact skyline probabilities based on relationships and probabilistic attributes of the graph. The last

skyline set contains all nodes that have the skyline probability higher than the stated threshold.

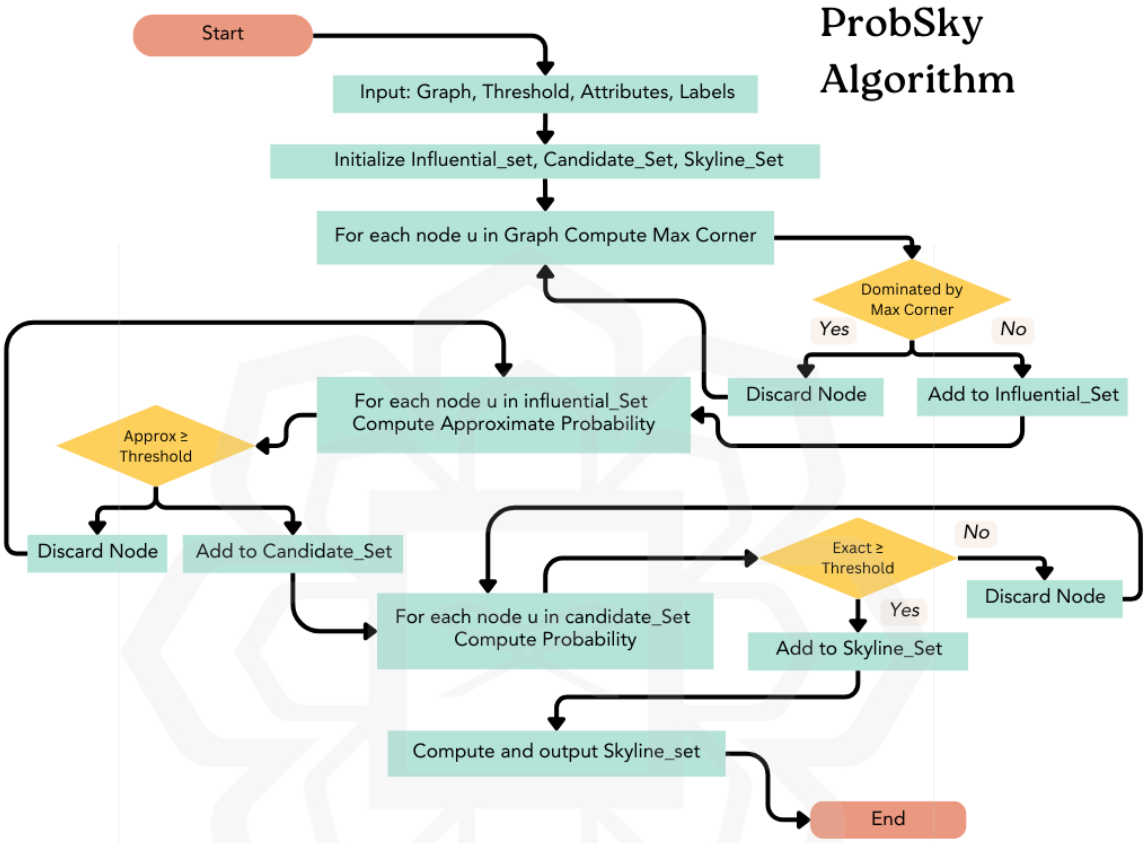


Figure 3.7 Flowchart for ProbSky Algorithm (Kuo et al., 2022)

In particular, ProbSky algorithm is more suitable to work with uncertain data's as it is designed to work with probabilities and connections. The pruning and refinement of the algorithm also make it well-suited for use with large databases, and therefore ideal for real-world applications in uncertain environments. With the help of local and global dominance relationships, the proposed ProbSky algorithm is able to construct a reliable algorithm for processing skyline queries in uncertain domain (Kuo et al., 2022).

3.6.2.3 U-Skyline Algorithm Overview

The U-Skyline algorithm also offers a potential solution to the skyline query problem in an uncertain dataset. It is an extension to deterministic skyline queries by taking into account the probabilistic characteristics of the data. The algorithm effectively manages probabilistic data through a recursive skyline probability computation for nodes, and it eliminates non-contributing instances to enhance the algorithm's performance. The pseudocode presented below shows an example of this adaptation for uncertain graph environments:

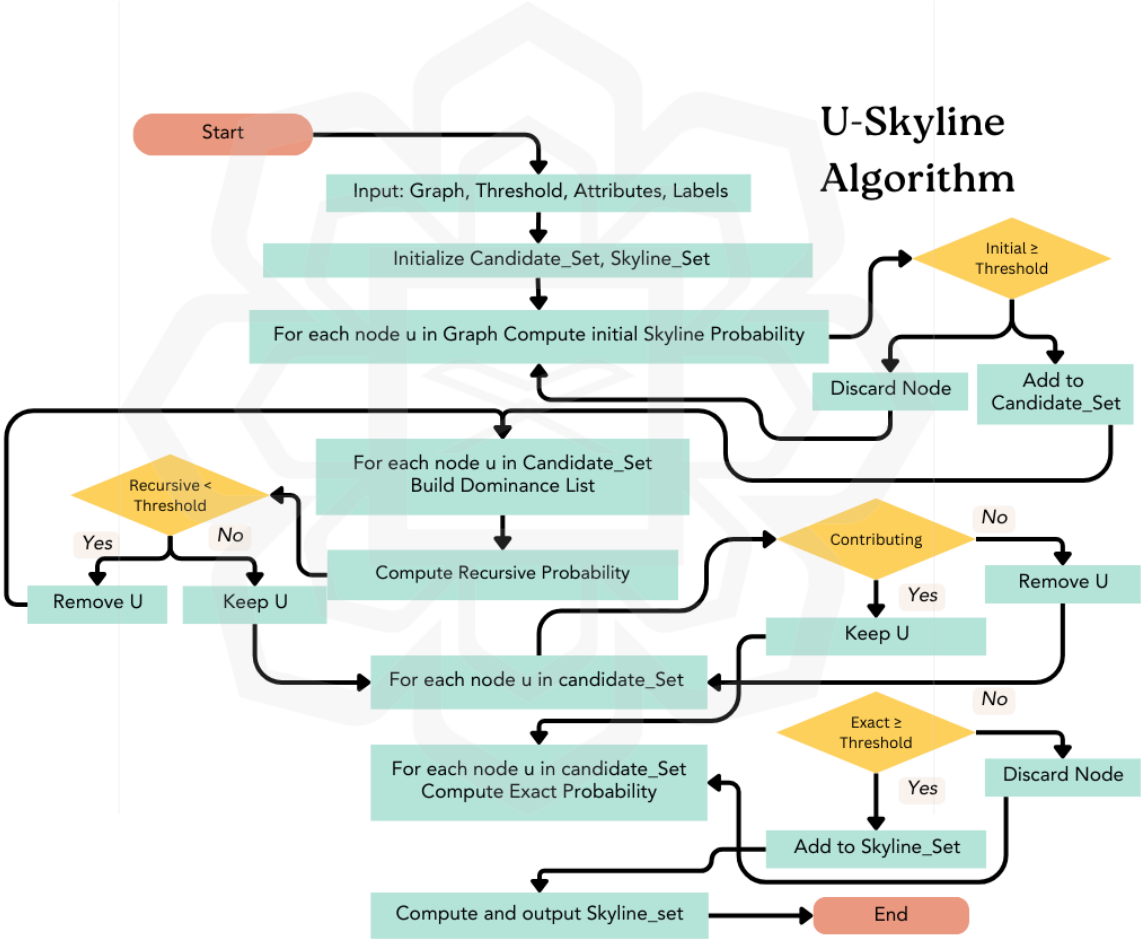


Figure 3.8 Flowchart for U-Skyline Algorithm (Liu et al., 2013)

The U-Skyline algorithm starts by setting candidate nodes. To decide whether a node belongs to the skyline set, each node is assessed concerning its probabilistic features. Non-

contributing nodes are eliminated based on a dominance check and the skyline probability of the other nodes is further computed iteratively. The last skyline set includes nodes for which the probability of skyline value is higher than a certain value. The algorithm updates skyline probabilities by considering both local information such as edge connectivity and global information such as dominance of one node over another in all attributes. This makes the algorithm efficient, and the results obtained accurate even when applied to large datasets. This is because the U-Skyline algorithm is best applied in situations where probabilistic attributes are relevant and key decision factors. For instance, in a transportation network, a node could be an intersection with ambiguous traffic conditions; an edge could be the connection between two intersections. The algorithm finds the critical intersections for the smooth traffic flow taking into account local and global uncertainty. Here, the U-Skyline algorithm includes pruning and recursive probability computation to guarantee scalability and efficiency. This is because an accurate skyline computation is required for decision making in large-scale uncertain datasets (Liu et al., 2013).

3.6.2.4 GNN and RL Algorithm

Integrating Graph Neural Networks (GNN) with Reinforcement Learning (RL) enables a strong algorithm for tasks on graphs that require decision making in an uncertain environment. In the case of uncertain graphs where nodes possess probabilistic attributes and where edges define dependencies or connectivity, this Hybrid Framework is suitable for skyline identification because it combines structural learning and sequential decision making. The GNN component encodes both structural and attribute information of the graph and the RL agent to dynamically update the skyline prediction. The RL agent communicates with the GNN and learns how to make better decisions to receive a positive reward signal that is derived from the GNN's predictions. This way the global graph features as well as uncertainty at the node level are incorporated in the skyline computation process.

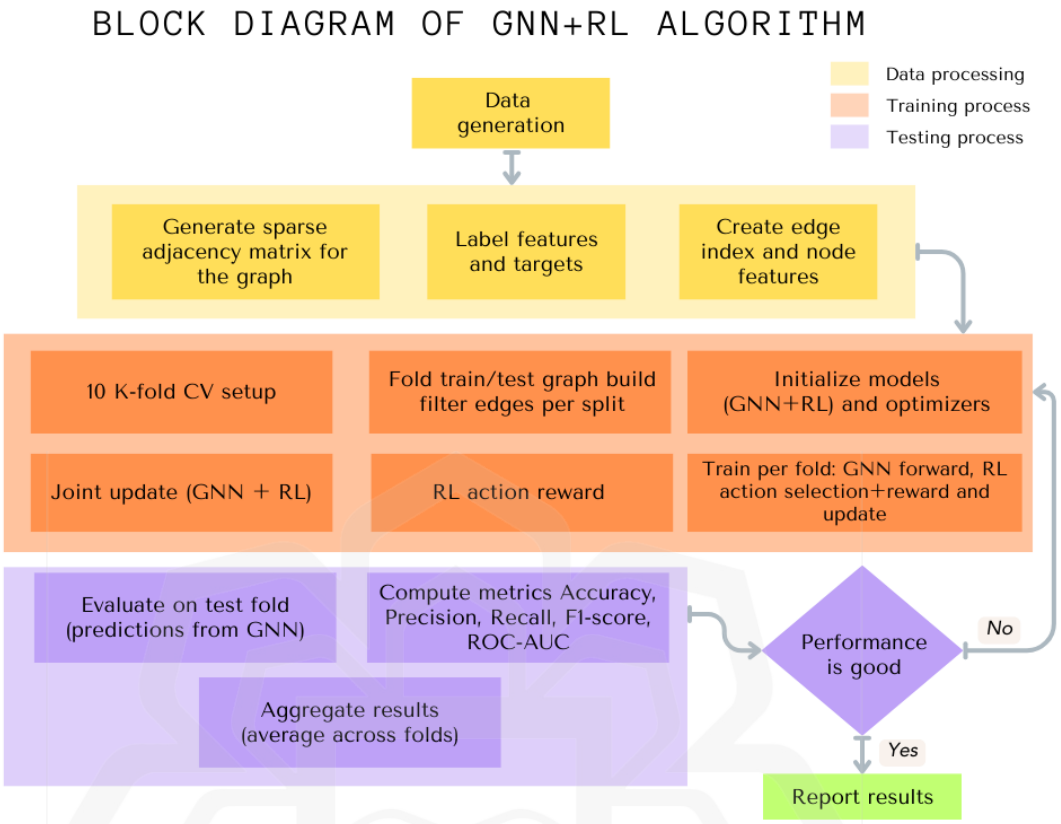


Figure 3.9 Block diagram for GNN and RL Framework

The GNN+RL algorithm starts by defining the graph structure, node features and labels. A GNN model is employed to propagate and aggregate the node features, to generate node representations. An RL agent takes these embeddings to make the skyline decisions. The agent is trained in policy gradient schemes where the reward function stems from the GNN's confidence in its decision. This creates a flexible and efficient algorithm for skyline query in large uncertain graphs to be used in real life scenarios like disaster management and urban planning.

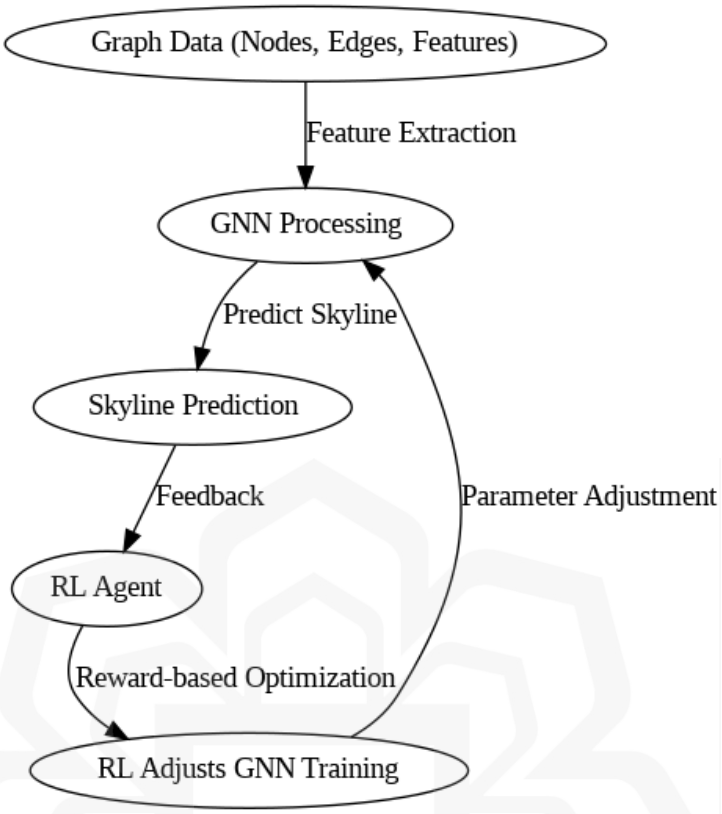


Figure 3.10 Hybrid workflow GNN and RL Framework

In the setting of skyline query processing, the diagram is the interaction between a Graph Neural Network (GNN) and a Reinforcement Learning (RL) agent. Instead of simply combining two independent methods, this hybrid framework uses the strengths of both GNN and RL by allowing RL agent to have direct effect on GNN training process. The flowchart clearly depicts the process by comparing it to different stages that build up the synergy between the two algorithms.

In the first stage, we have the Graph Data (Nodes, Edges, Features) that is the basis of entire algorithm. It is a dataset of a network of entities with nodes representing objects of interest (e.g. locations in a transportation network) and edges describing relationships between them. Additionally, for each edge there are multiple attributes, for instance, distance, travel time, congestion level as well as traversal probability. After that the GNN

Processing is started. However, those GNNs that are used to extract meaningful patterns from the graph structure and its associated features. GNN is used to take advantage of the node dependencies and relationships to cope with structured data such as graphs. Given the graph data, the GNN performs message passing mechanism to learn the node and edge representations in this step. These representations encode useful information about each node's connectivity, feature distribution, and relevance to the skyline query. The initial prediction of which nodes belong to the skyline set are generated using the learned representations.

After GNN processing, the model does Skyline Prediction. At this point, the GNN selects a set of candidate nodes which it believes to be the skyline points according to multi criteria optimization. According to the current training parameters, these are the nodes which offer the best trade-off between the attributes defined by the training parameters. However, this begins to reveal more of the hybrid nature of the algorithm. This method is different than a traditional GNN based algorithm which intended for the selection predictions to only be made based on learned embeddings and loss minimization. In this algorithms the selection process is further refined by an RL agent. To refine the skyline predictions, the system presents the RL Agent which assesses the quality of GNN produced predictions. Unlike GNN, the RL agent does not trust on its outputs passively; the agent executes actions after considering the outputs and providing feedback utilizing a reward mechanism. The RL agent turns the feedback loop essential: the agent is not dependent solely on the GNN's training loss function, while determining whether the skyline predictions match an optimal decision-making strategy. Correct predictions are given rewards by the agent while incorrect ones incur penalties and as a result the GNN adapts to real world constraints and dynamic conditions.

In the next step, RL Adjusts GNN Training, this method deviates from simple combination of GNN and RL by conducting core hybridization that occurs in RL. In this framework, the RL agent changes the GNN's training process actively itself which means that it alters the mechanism through which the GNN learns the representations in the first

place. Finally, this step updates the GNN's parameters by using the reward signals received from the RL agent. When some skyline predictions are always rewarded by the landscape, the GNN learn those patterns and reinforce it using its internal weights, and when other predictions are punished by the landscape this becomes accounted for by the model's learning and recalibrates.

This is the key difference between a hybrid and a combination algorithm, because of how they interact. In a combination algorithm, the two models operate independently, with their outputs being merged at a later stage. In that case, the RL agent is placed inside the training loop of the GNN so that the learning trajectory of the RL agent can influence the learning trajectory of the GNN. This is why the method, as a hybrid and not just a combination of the two, sees how the two techniques work side by side, while the RL agent controls the GNN training.

3.7 Performance Evaluation

3.7.1 Key Performance Metrics for the experiment

Efficient skyline query algorithms in complex and large-scale datasets such as uncertain graph databases demand thorough considerations in multiple performance metrics. The different perspectives of these metrics provide evidence that even when the skyline query algorithm correctly identifies the skyline points, the false positives and false negatives are minimized. In this research, the Algorithms are evaluated using key metrics such as Accuracy, Precision, Recall, F1 Score, and ROC-AUC. These metrics are equally important to the algorithm's performance, especially in the case of imbalanced data and each of these metrics determines the algorithm's overall performance.

Comparison to accuracy is the most basic and commonly used metric in classification problems for which accuracy is defined to be the ratio of correctly classified points (skyline and non-skyline) over the total number of points (Vujovic, 2021). This is just a rough estimate of how much the algorithm is working for all the classes. However, accuracy is misleading in the worst case when the datasets have a very skewed class structure, like skyline queries. The number of non-sky line points is usually much more than the number of skyline points in skyline query processing.

Precision represents the ratio between the true positives (rightly identified skyline points) among all classified as skyline points (true positives and false positives) (Yacouby & Axman, 2020). The relevance to the output of these skyline queries is important because of this metric. Having a high precision score indicates that the algorithm is not flooding the skyline result set with non-relevant (non-skyline) points. In applications in which decision-makers rely on skyline points to make the optimal decisions, precision ensures that the output contains only truly optimal or near-optimal choices which means decreasing the noise in the output.

The term recall talks about the proportion or true positive count over the total number of actual skyline points (Powers, 2020). In cases where missing skyline point points could cause very significant consequences, for example in logistics, transportation, and recommendation systems, this metric is very important. A high recall score means the algorithm finds most of the true skyline points, and thus gets completeness in the query result.

As a balance metric between precision and recall, the F1 score is the harmonic mean of the two (Mortaz, 2020). It is useful in skewed datasets where we only have to balance precision vs recall.

Another important metric in this research to evaluate the discriminative power of skyline query algorithms is ROC-AUC. ROC curve assigns a true positive rate (recall) on the y-axis and a false positive rate on the x-axis as it visualizes the tradeoff between sensitivity and specificity as classification threshold varies over different classification thresholds (Carrington et al., 2022). The AUC is the area under this curve, and so represents how discriminative this curve is, and a higher value of the AUC indicates better discriminative ability. In the skyline query setting a high ROC AUC score means that the algorithm can better separate skyline and non-skyline points regardless of its decision threshold. This is particularly important in an uncertain environment where the algorithm may require those changes to deal with dynamic data distributions.

3.7.2 Cross-Validation and Experimental Setup

In this research, K-Fold Cross-Validation was employed as the primary evaluation methodology to make sure that the performance metrics faithfully summarize the generalizability of skyline query algorithms. Partitioning the dataset consists of training the model using a subset of said 'fold' of the dataset, and testing on the remaining fold (test), and the method is called "k-fold", where k is the number of folds in the data. It’s repeated multiple times, and each fold is just run as the test set exactly once. Then its averaged results from each fold are used to yield a more reliable estimate of the model's performance.

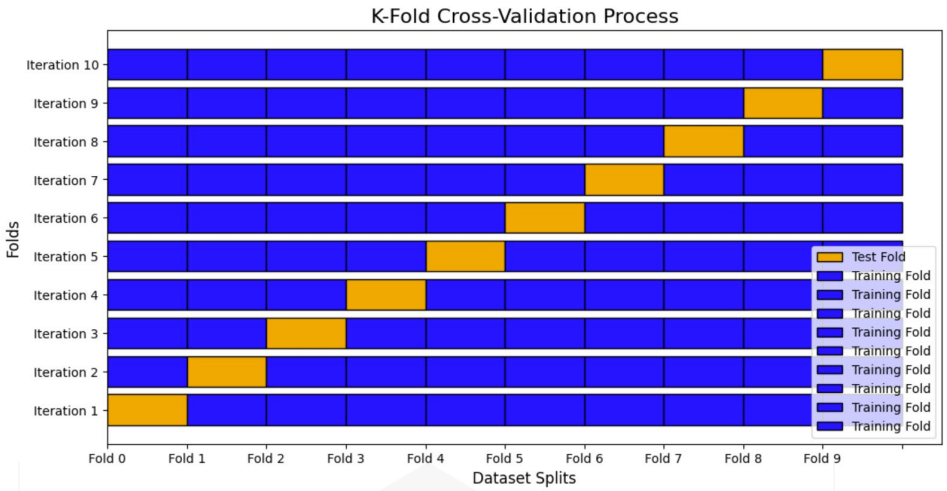


Figure 3.11 K-Fold Cross-Validation Process

Cross-validation use of K is a crucial parameter. In this research, K = 10 was chosen so that the dataset was divided into 10 equal-sized folds. It was trained on 9 folds and tested on the unseen fold, and we repeated this 10 times, with each fold being the test set once. And here researchers chose 10 folds to strike a balance between computational efficiency and robustness. In addition, a random seed equal to 42 was used to make the cross-validation reproducible. This will split data into folds in such a way that the data is split into folds using the random seed and thus the data should remain the same, provided the experiment is repeated the same, for different runs of the experiment. For skyline query processing specifically, in imbalanced datasets where skyline points are rare, small changes in the data distribution can greatly affect the algorithm's performance, making this problem especially important (Zhang & Liu, 2023).

In addition, oversampling of the minority class (skyline points) in each fold was performed as the cross-validation setup, to address the class imbalance problem. Skyline points are typically a very small subset of the dataset, so we need to handle this well to avoid getting biased toward the majority class (non-skyline points). To mitigate this, the RandomOverSampler method was applied to each training fold to ensure that the number of skyline points is balanced with the number of non-skyline points.

3.8 Summary

This chapter outlines the methodology used to evaluate skyline query processing in uncertain graph databases. The research follows a structured approach, starting with data preparation and handling class imbalance using Random Oversampling to ensure a fair representation of skyline points. To assess algorithm performance, 10-fold cross-validation was employed, providing a robust and generalizable evaluation by splitting the dataset into multiple training and test sets.

The performance evaluation framework was designed using key metrics, including accuracy, precision, recall, F1 score, and ROC-AUC, ensuring a comprehensive assessment of each algorithm. The choice of these metrics reflects the need to balance false positives and false negatives, especially given the imbalanced nature of skyline queries. The methodology ensures that the evaluation process is reliable and replicable, providing meaningful insights into the effectiveness of different skyline query algorithms in uncertain environments.

CHAPTER FOUR

PRELIMINARY FINDINGS

4.1 Introduction

This chapter presents the preliminary experimental findings obtained from evaluating skyline query processing algorithms on a synthetic graph dataset. The purpose of this chapter is to provide an initial comparative assessment of baseline skyline algorithms and learning-based approaches using standard classification performance metrics. These results serve to establish baseline behavioral patterns and highlight early performance differences among the evaluated methods. The findings presented in this chapter are intended to be exploratory and descriptive in nature. They provide foundational insights that motivate the more comprehensive analysis, stress testing, and in-depth discussion of scalability, robustness, and multi-objective quality evaluation.

4.2 Experimental Results

4.2.1 Top-K Skyline Objects Algorithm

The Top-K Skyline Objects Algorithm is a deterministic algorithm for finding the Top K skyline objects from a dataset. The algorithm works with a selection of top K data points that best satisfy dominance conditions by ranking data points according to defined criteria. A point dominates another if it's at least as good in all dimensions, and strictly better in at least one. Second, this algorithm aims to minimize the size of the skyline as it focuses on the most important points to minimize the problem of skyline overpopulation which is common in high-dimensional data (Sukhwani et al., 2021).

Practically, the Top-K algorithm generates the complete skyline and then ranks the skyline points on the total score across attributes. The order of this ranking can be tailored to the application, with certain attributes being more weighted than others. This algorithm is useful in recommendation systems where users are interested only in the top few best results.

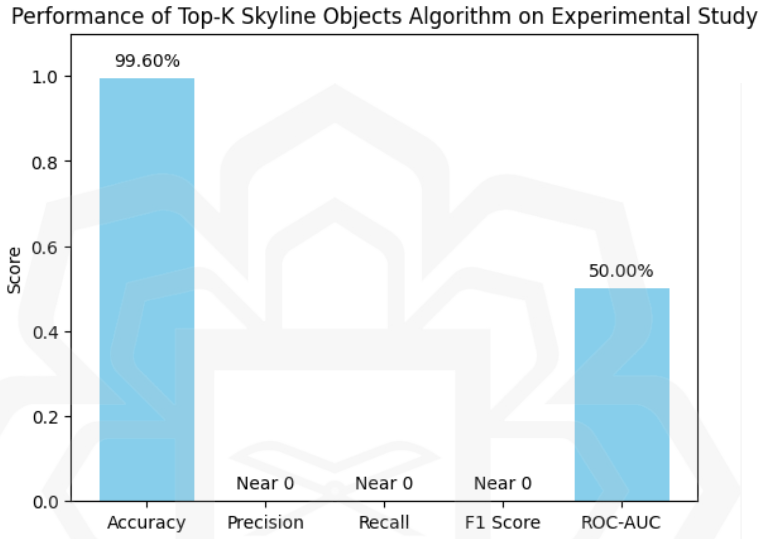


Figure 4.1 Performance of Top-K Algorithm on experimental research

Nevertheless, the experimental results from this research pointed out several limitations in the Top-K Skyline Objects Algorithm. While this achieved high accuracy (0.9960), precision, recall, and the F1 scores were near zero due to the poor ability of the algorithm to identify the minority class which are the skyline points. It suggests that the high prevalence of dominated points and how the algorithm handles high dimensional data prevented the algorithm from identifying the correct skyline points consistently and led to good performance for the non-skyline points identification.

4.2.2 ProbSky Algorithm

ProSky algorithm addresses the probabilistic skyline query computing for distributed data systems, assuming the attributes of each point are probabilistic. This algorithm is useful for spaces that have uncertain data points, such as transportation networks where travel times, distances, and costs differ based on external conditions such as traffic or weather. ProSky algorithm, instead of finding a fixed set of skyline points, determines the probability that a given point is in the skyline, allowing for a more flexible query pattern. The key steps involved in the ProSky algorithm include: Modeling Uncertainty, Probabilistic Dominance and Skyline Query (Kuo et al., 2022).

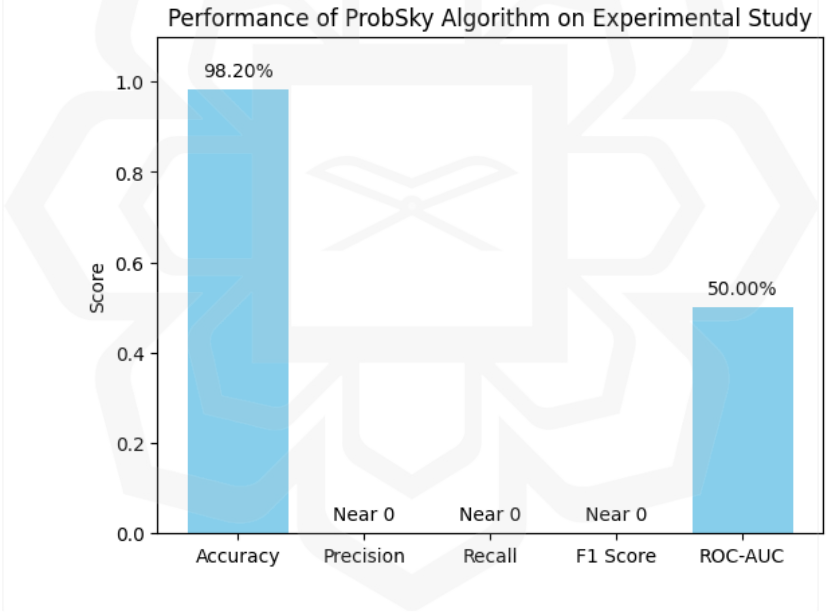


Figure 4.2 Performance of ProbSky Algorithm on experimental research

However, the experimental results of this research showed that large datasets and uncertainty are major limitations of the ProSky algorithm. Although the algorithm performed well with a high accuracy of 0.9820, it did not perform well in terms of precision, recall, and F1 score, which were all near zero, showing that the algorithm did not consistently identify the skyline points. The source of this issue could also be that the

algorithm does not scale well with the size of the dataset, or that it is difficult to scale with complexity due to probabilistic dominance comparison in the dimensions space.

4.2.3 U-Skyline Algorithm

The U-Skyline algorithm proposes a algorithm to answer skyline queries on uncertain graph databases. This algorithm is able to account for the fact that in many real-world networks relationships between entities for example, connections in a transportation network are not constant but rather a function of probabilistic factors, such as road closures, traffic delays, or variable costs (Liu et al., 2013). The uncertain node attributes and probabilistic edges are modeled and probability is taken to compute the likelihood that given a particular node will form part of the skyline. The algorithm generalizes the notion of dominating to uncertain dominating, defining that one node probabilistically dominates another because it is likely to have all dimensions of better attribute values.

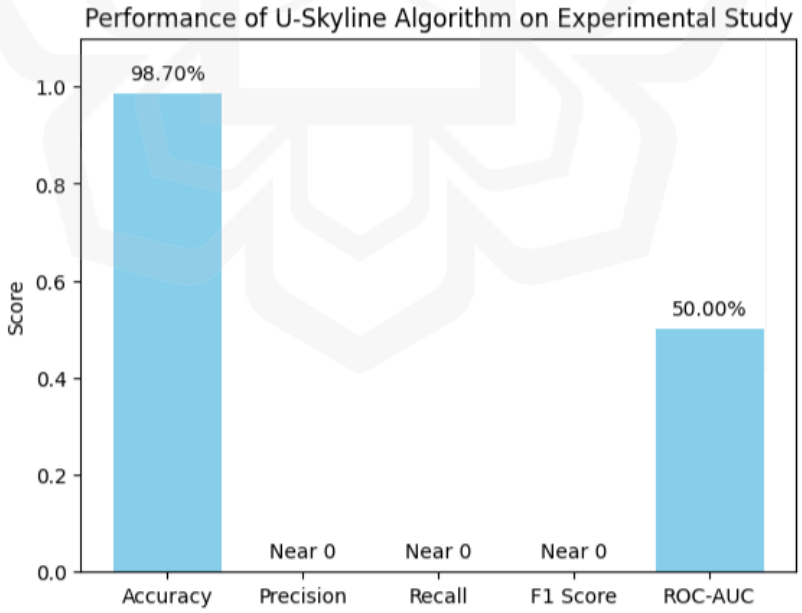


Figure 4.3 Performance of U-Skyline Algorithm on experimental research

The experimental results of this research indicate that the theoretical advantages of handling uncertainty in the U-Skyline algorithm have yet to be turned into practice in large scale data. While the algorithm achieved a very high accuracy of 0.987, precision, recall and F1 score were all near zero, indicating that the actual skyline points were not identified correctly. These results show that the U-Skyline algorithm is not as scalable to large datasets, nor will it compute skyline queries efficiently when uncertainty is high.

To summarize, the U-Skyline algorithm provides a novel means to address uncertainty and its empirical shortcomings of low precision and recall suggests that further optimization is required to make it suitable for real-world applications in very large, uncertain graph databases.

4.2.4 GNN Based Algorithms

Graph Neural Networks (GNN) are a natural fit for processing skyline queries in large-scale and uncertain graph databases because they are specifically designed to process graph-structured data. Unlike pairwise comparison-based algorithms, which find skyline points by comparing each entity with every other possible entity, GNN leverages the graph structure of nodes and edges to model complex relationships and dependencies among entities in a scalable and efficient way.

In this work, researchers use graph neural network model architecture with multiple layers of graph convolutional networks, called GCN. On specific sorts of graphs, a GCN is a GNN based on convolution operations focused on combining and spreading information between correlative nodes. This algorithm enables learning of better node embeddings for the model in an iterative fashion updating each node's representation with the features of its neighbors and the graph structure as a whole. Here, the model used has two graph convolutional layers coupled with fully connected layers, batch normalization, and dropout

layers to avoid overfitting. The GNN layers transform a set of features for each node: distance, probability, travel time, congestion levels, etc., and learn them.

The reason that the GNN is particularly well suited for large graph databases, is that its representation of each node depends on the features of neighbors of that node. GNN can scale efficiently for large datasets, by performing this localized computation, without having to do exhaustive pairwise comparisons which is the case with existing skyline algorithms. Furthermore, GNNs are readily adapted to uncertainty and can learn to model probabilistic relationships between nodes while simultaneously capturing uncertainty in node attributes and the connections between them.

Researchers of this thesis applied the GNN to a synthetic dataset of 5,000 nodes and 250,000 edges, where each node and edge was represented in a set of features related to skyline query processing. They trained the model as a supervised learning problem based on feature and connection information where the aim was to classify nodes as either skyline points or non-skyline points. Results of the GNN model demonstrated substantial improvements compared to baseline skyline query algorithms, especially concerning its capacity for handling uncertainty and scalability.

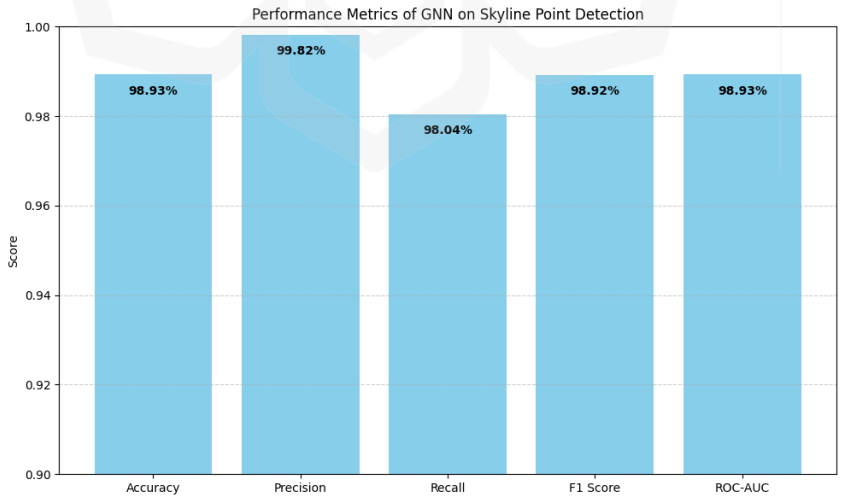


Figure 4.4 Performance Metrics of GNN

Key metrics such as accuracy, precision, recall, F1 score, and ROC-AUC were used to evaluate the performance of the GNN model. In particular, the GNN achieved an average accuracy of 0.9893, a precision of 0.9982, a recall of 0.9804, and an F1 score of 0.9892 with a ROC-AUC of 0.9893.

Table 4.1 Performance Metrics of GNN on Skyline Point Detection

GNN	
Accuracy	98.93%
Precision	99.82%
Recall	98.04%
F1 score	98.92%
ROC-AU	98.93%

In contrast with Top-K, ProSky, and U-Skyline algorithms which have shown low precision and recall, the GNN was able to find the king of the mountain by providing true skyline points while avoiding false positives. GNN manages to propagate information across connected nodes, which is critical to skyline queries in graph databases as it allows to capture of complex relationships that happen between pairs of nodes. Additionally, the GNN proves to be a powerful tool for skyline query processing in dynamic environments where a skyline may vary with the changing nature of the data, the uncertainty in the data, or large datasets over time and across operating domains

4.2.5 GNN and RL Based Algorithms

A hybrid GNN + Reinforcement Learning (RL) framework was further developed to optimize skyline query processing. In this framework, a GNN component learns from the graph structure and node representation, while the RL component enforces dynamic

decision-making to optimize the efficiency and effectiveness of the skyline query process further. This integration of RL allows the framework to learn policies for node selection that steer towards evaluation, and perform skyline detection in a more targeted manner, without relying on exhaustive comparisons.

The RL component applies the policy gradient algorithm, whereby an agent learns to take actions like select the node(s) to evaluate, depending on the state of the representations learned by the GNN. Rewards are leveraged to update the policy taken from the GNN output. To drive such a process, if the GNN classifies the node as a skyline point correctly, then the agent gets a positive reward to reinforce the decision-making process. The RL agent will intermittently prioritize nodes that have a higher propensity of being skyline points, and by so do so, improving the efficiency of the query process, as the RL agent learns over time.

Researchers of this thesis introduce the RL component that has several advantages over using GNN alone. The RL agent then reduces the complexity of evaluating every node of the graph by mastering the ability to spend time only with the most promising candidates. The second advantage of the RL algorithm is that the framework can be adapted dynamically over time as node attributes or node relationships become more uncertain. Third, the policy-based decision-making process helps improve the overall query efficiency, especially in large-scale databases where the computation of exhaustive skyline queries is prohibitively costly.

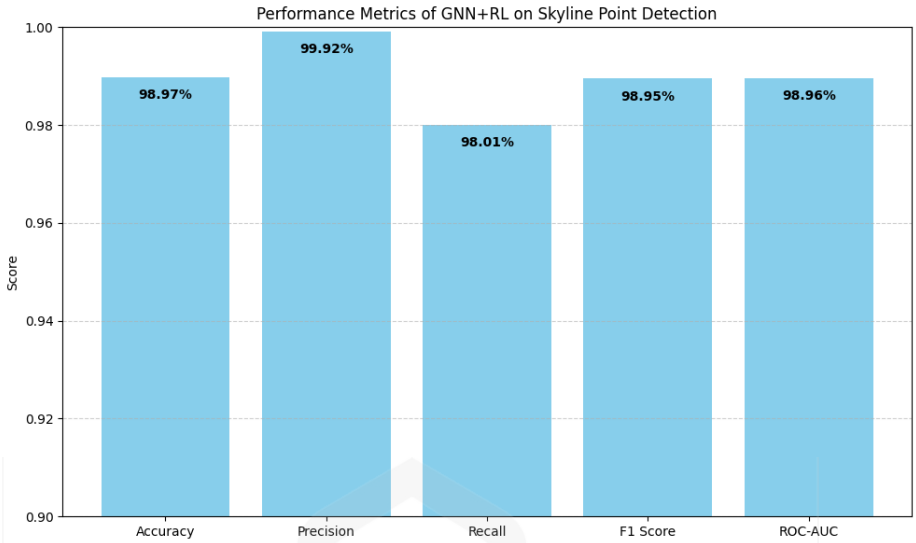


Figure 4.5 Performance Metrics of GNN+RL

Researchers presented the performance results of the hybrid GNN + RL framework displaying even greater improvement over both baseline skyline algorithms and the standalone GNN model. In the hybrid framework, used above, the average accuracy achieved was 0.9897, precision 0.9992, recall 0.9801 and F1 score 0.9895, and ROC-AUC 0.9896.

Table 4.2 Performance Metrics of GNN+RL on Skyline Point Detection

GNN + RL	
Accuracy	98.97%
Precision	99.92%
Recall	98.01%
F1 score	98.95%
ROC-AU	98.96%

The results show that RL integration led to increases in the precision score and usually the query efficiency, reflected in the near-perfect precision score. The hybrid

framework is highly effective for skyline query detection in uncertain graph databases because by filtering the most likely skyline points in the query process and relying on an RL agent to help eliminate false positives, both recall and false positive rates were reduced.

This work introduces the GNN+RL framework, which brings a novel optimization algorithm to skyline query based on the structural learning capabilities of GNN and the decision-making capability of RL and demonstrates dramatically improved optimization performance compared to baseline algorithms.

In particular, the hybrid framework is highly useful for dynamic and uncertain environments where the graph structure and attributes are likely to change over time. The ability to adapt to these conditions makes it a candidate solution for real-world use, such as optimization of the routes in the transportation network or key products in e-commerce recommendation systems.

4.3 Research Insights and Challenges

Despite the key success of all the listed algorithms to identify skyline points, there are major challenges for baseline skyline algorithms in terms of precision, recall, and F1 score, and these algorithms face significant inherent limitations in complex and uncertain environments. Even if these algorithms such as Top-K, ProSky, and U-Skyline reliably compute points, they do not do well at giving certainty to a skyline point.

Baseline algorithms have a dependence on deterministic dominance comparisons, where a point is deemed superior over another point if the first is at least as good as the second in all dimensions, and strictly better in at least one. This becomes problematic when the points' attributes vary and dominant relationships are unclear, in scenarios with

uncertainty. For that reason, these algorithms tend to misclassify skyline points and achieve very low precision and recall scores. In practice, these results mean that decision makers relying on them may overlook important optimal choices and thus result in suboptimal outcomes.

Additionally, the class imbalance problem crops up in skyline queries where the number of skyline points is usually small when compared to the total dataset. Such imbalances can lead to skewed performance of baseline algorithms, particularly when the majority class (non-skyline points) are targeted, raising the challenge of designing algorithms to accurately identify skyline points. The results here were consistent with this imbalance as the performance metrics of the baseline algorithms were able to achieve high overall accuracy while failing to obtain meaningful precision and recall scores.

It also becomes especially important in this situation, the F1 score being that which balances precision and recall. Previous baseline algorithms reported an F1 score of near zero, meaning that they could not detect any skyline points while being extremely accurate. This brings to light the necessity for skyline algorithms to identify points accurately and to obtain skyline points whose results can shape a decision-making process. Doing so will ultimately render skyline queries ineffective in the presence of real-world applications where relevant and reliable results are needed. However, the GNN and the GNN + RL frameworks address these challenges better. GNN takes advantage of the connectivity defined by the inherent structure of graph data to make finer distinctions between points. GNN then propagates information across connected nodes to learn richer representations of points that enable them to determine skyline points more accurately. All this structural learning capability enables GNN to be robust in the presence of uncertainties, which baseline algorithms are unable to handle. Also, combining GNN + RL leads to a dynamic decoding process with reinforcement learning (RL) that is flexible to fluctuating data conditions. During the training process, feedback is received by the RL agent, and it will learn to give some priority to some nodes that are more likely to be skyline points. Such adaptability not only improves the precision and recall scores of the framework but also

helps improve overall query efficiency, such that skyline queries may still return relevant answers in complicated and uncertain datasets.

4.4 Summary

In this chapter, researchers of this thesis compare baseline skyline algorithms (Top-K, ProSky, and U-Skyline) with modern algorithms, e.g., Graph Neural Networks (GNN) and a hybrid GNN + Reinforcement Learning (RL) framework. Experimental results show that, despite high accuracy, baseline algorithms were unable to identify a small number of key points of the skyline, and were consequently characterized by a low precision, recall, and F1 score. These results uncover a huge bottleneck in the ability of baseline skyline algorithms to deliver accurate identification of optimal solutions in practical settings.

On the other hand, the GNN and GNN + RL frameworks exhibited excellent performance in regard to all of the key metrics. As the GNN model can learn the structure of a graph and propagate helpful information in node-connected nodes, it could successfully identify skyline points with a high precision and recall rate. Further enhancement was achieved by using the hybrid GNN + RL framework. This chapter's findings emphasize the critical need to utilize cutting-edge deep learning methodologies, such as GNN and reinforcement learning, to tackle the issues of skyline query processing in complex and high dimensionality data.

CHAPTER FIVE

RESULT AND DISCUSSION

5.1 Introduction

Thorough testing is necessary before proposing any algorithm to determine its value in both theory and practice. Here, researchers endeavor to verify how their proposed Graph Neural Network (GNN) and Reinforcement Learning (RL) algorithm perform using a wide range of tests in uncertain graph skyline queries. Researchers plan to verify the accuracy of the GNN+RL framework by running it under various stress tests on different graph scenarios and their corresponding densities. Researchers also strive to define the boundaries of the algorithm in terms of performance, its response to disturbed data, and the effects of varying thresholds.

5.1.1 Purpose and Scope of Testing

While regular approaches for skyline data use values that do not vary, researchers must account for uncertain edge existences and their range of possible values. For this reason, researchers organized this testing to include architectural experiments to determine the effects of normalization and regularization, as well as scalability and performance indices, to demonstrate how well the algorithm predicts. With this group of categories, researchers can thoroughly analyze every aspect of the system.

5.1.2 Details of the stress-Test categories

This test is divided into four major sections. In the first section, experiments without either Batch Normalization or Dropout reveal their role in aiding training and preventing overfitting. For the second category, the recorded details include execution time and maximum memory required, as these depend on the number and type of nodes, as well as the number of edges in the graph. Then, researchers check the framework's accuracy, precision, recall, F1-score, and ROC-AUC while setting the ground truth value (skyline points) at the 90th percentile and above.

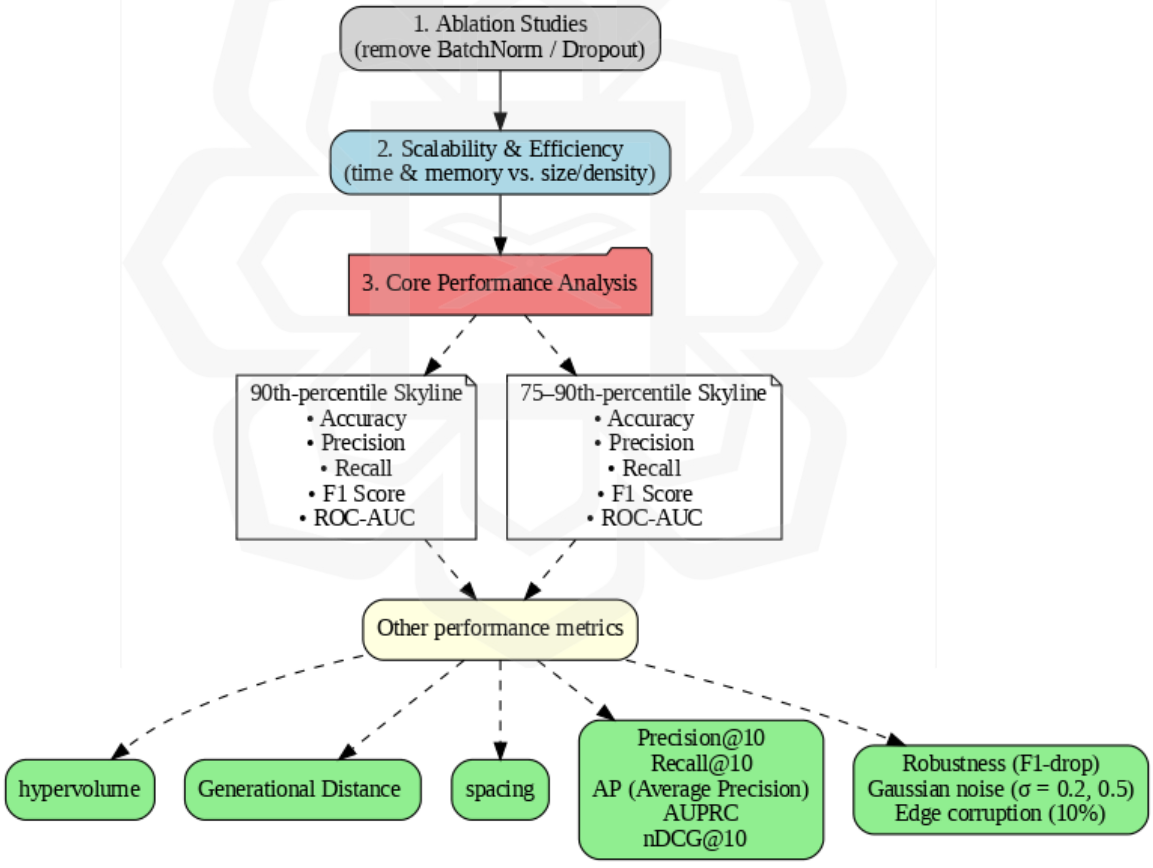


Figure 5.1: Experimental test setup

Alongside the core metrics, hypervolume, generational distance, spacing, Precision@10, Average Precision (AP), AUPRC, and nDCG@10 are used to measure and evaluate how

well the framework handles the true skyline and ranking list. In addition, structural changes and noise have been introduced to assess the limitations of the proposed framework. However, defining the skyline points or ground truth at the 90th percentile creates a textbook-defined skyline, which is very easy to predict using an advanced framework like GNN+RL. To ensure the test results are relevant, researchers have also tested the algorithm by randomizing the ground truth to range from the 70th to the 90^{th} percentile.

5.2 Experimental Setup

5.2.1 Hardware and Software

The experiments were carried out using a Mac Mini M4 and a Ryzen 7 laptop. This Mac Mini features an Apple M4 chip, supports a neural engine with eight cores, and has sixteen gigabytes of unified memory, all operating the latest version of macOS. The laptop features an 8-core/16-thread processor, 32 GB of RAM, and an NVIDIA GeForce GTX 1660 Ti graphics card. Additionally, Google Colab has been utilized under certain conditions where local machines failed to calculate due to complexity.

Python 3.13.3 was used to execute the code. Libraries such as PyTorch and PyTorch Geometric were utilized to implement the GNN and RL modules. To manipulate the baseline graph, researchers used a library named NetworkX. For working with the data, several modules, including numpy, pandas, sklearn, and imblearn libraries, were utilized. Here, F1-score, ROC-AUC, AP, AUPRC, nDCG, and similar metrics were calculated with sklearn.

5.2.2 Implementation of GNN+RL Framework test

The researchers proposed a hybrid architecture that seamlessly integrates a Graph Neural Network (GNN) with a Reinforcement Learning (RL) policy module to address skyline query processing under uncertainty effectively. The experimental framework is designed to simulate real-world decision-making scenarios using synthetic graph data, where node features are first annotated and enhanced with skyline-relevant semantics. These enriched features are subsequently aggregated and propagated through a GNN-based encoder, facilitating structural awareness and context-sensitive embeddings.

To systematically evaluate the algorithm across a range of conditions, the researchers generate synthetic uncertain graphs following an Erdős–Rényi random model. Node counts are varied among $N \in \{5{,}000; 10{,}000; 20{,}000; 50{,}000\}$, while edge densities are set at $d \in \{1\%; 2\%; 5\%; 10\%\}$. In each graph realization, every potential edge is assigned an existence probability drawn uniformly from the interval [0.5, 1.0], capturing scenarios ranging from moderate to highly reliable connectivity. For each graph instance, the researchers set aside 0.2% of the graph size as Skyline points or ground truth. These dominance counts form an empirical distribution from which percentile thresholds are defined: initially at the 90th percentile to research extreme-threshold behavior, and subsequently over the 75th–90th percentile range to examine more nuanced discrimination tasks. This rigorous data-generation protocol ensures uniform coverage of network scales and controlled variation in uncertainty levels, enabling comprehensive analysis of the framework's strengths and limitations.

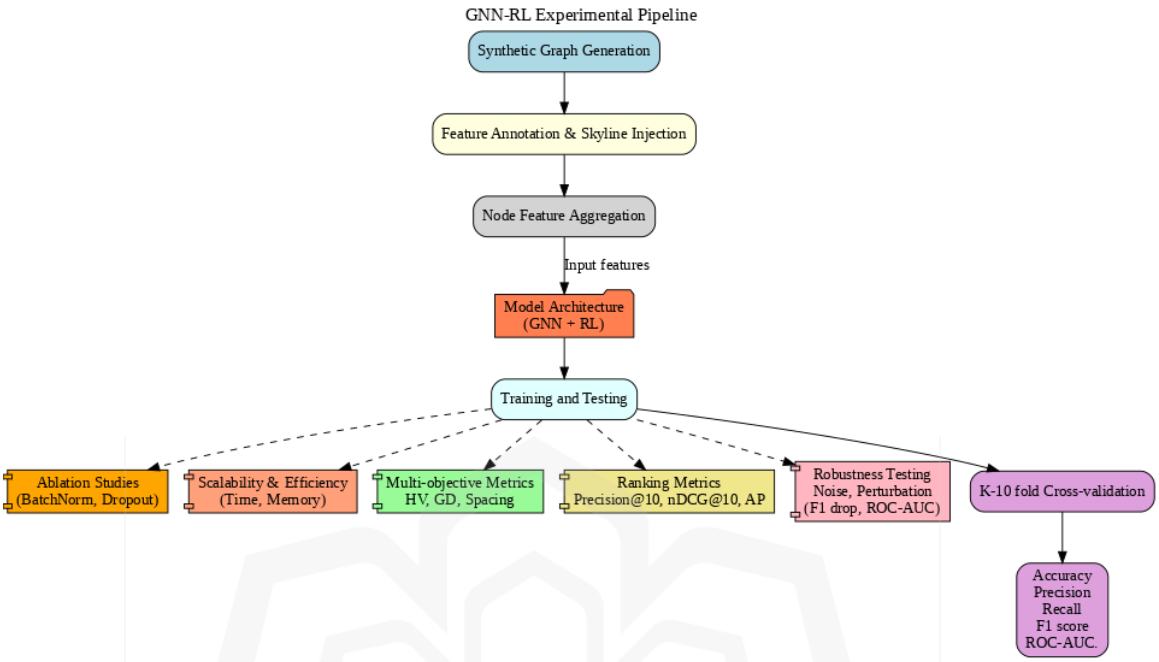


Figure 5.2: GNN+RL experimental setup pipeline

At the core of the architecture lies a reinforcement learning module that operates on top of the GNN representations, optimizing selection strategies over the multi-objective trade-offs inherent in skyline computations. The training phase is carefully designed to incorporate feedback from both scalar performance signals and ranking-based evaluations, enabling the framework to learn not just to classify but to prioritize relevant results.

To rigorously validate the proposed framework, the researchers conducted a diverse set of evaluations encompassing ablation studies, scalability analyses, and robustness checks against adversarial noise and graph perturbations. For core classification, the researchers record accuracy, precision, recall, F1-score, and the ROC-AUC, thereby capturing both threshold-based and ranking-based discriminatory power. To evaluate the quality of multi-objective approximations relative to the true Pareto front, researchers compute hypervolume (HV), generational distance (GD), and spacing (Sp), which respectively measure coverage of the objective space, average proximity to the front, and uniformity of point distribution. Complementing these, ranking fidelity indicators Precision@10,

Recall@10, Average Precision (AP), area under the precision–recall curve (AUPRC), and normalized discounted cumulative gain at ten (nDCG@10) provide insight into how well the framework prioritizes the most critical skyline candidates. Finally, robustness is assessed via the F1-drop observed when additive Gaussian noise ($\sigma = 0.2$ and $\sigma = 0.5$) is introduced to node features alongside the random drop of 10% of edges. Collectively, these metrics furnish a holistic perspective on predictive accuracy, Pareto-approximation fidelity, ranking quality, and resilience to perturbations.

5.3 Dataset

In this research, the synthetic dataset was generated with an aim to simulate large-scale uncertain graph databases reflecting realistic conditions in transportation and logistics networks. The data generation process follows a controlled but flexible approach that allows systematic variation of graph size, edge density, and noise levels to evaluate the performance, robustness, and scalability of skyline query algorithms.

The graph is constructed using an Erdős–Rényi random model, where nodes represent entities (e.g., locations), and edges represent probabilistic relationships or routes between nodes. The node counts NNN are varied across experiments, with typical values including 5,000, 10,000, and 20,000 nodes to assess scalability. Edge density ddd is specified as the ratio of existing edges to the total possible number of edges, with values tested ranging from 0.005 to 0.05 (0.5% to 5%) and beyond in stress testing. For each edge, multiple features are generated to capture realistic multi-dimensional attributes affecting skyline computations.

Edge features include:

- **Distance:** An integer value representing the physical or temporal distance between the source and target nodes. It is randomly sampled from 1 to 100.

- **Probability:** A floating-point value representing the likelihood of successfully traversing the edge, drawn from a uniform distribution between 0.5 and 1.0, modeling uncertainty in edge existence.
- **Travel Time:** An integer representing estimated travel time for traversing the edge, generated between 1 and 120 and correlated with distance, with noise to simulate real-world variability.
- **Congestion:** An integer depicting traffic congestion levels, sampled from 1 to 10, modeling traffic impact.
- **Weather Impact:** An integer score from 1 to 5 representing the influence of weather conditions on the edge usability.
- **Correlated Feature:** A derived floating-point attribute constructed as the sum of Distance and Travel Time with added Gaussian noise, simulating complex correlations among features.

To generate node features suitable for graph neural network inputs, edge attributes are aggregated for each node by computing the mean of all originating edges' features. To better simulate real-world noise and measurement uncertainty, Gaussian noise proportional to graph size is added to these aggregated node features.

The synthetic dataset further assigns skyline labels to nodes, identifying approximately 0.2% of nodes as skyline points. These nodes are randomly selected and assigned marginally improved edge attributes specifically, their associated distances and travel times are reduced by 10%, and probabilities increased by 10% to represent optimal or near-optimal points in the dataset.

The selected attributes directly influence the dominance relations central to skyline queries. Skyline points represent nodes that are not dominated by any other node across all

considered attributes. In this dataset, attributes such as Distance, Travel Time, and Probability define multi-objective criteria where a node is preferred if it has lower distances and travel times while maintaining higher traversal probabilities. Congestion and Weather Impact further affect the overall desirability and feasibility of paths associated with each node. The inclusion of a correlated feature captures realistic dependencies among attributes, adding complexity to the skyline decision boundary. By marginally improving these attributes for designated skyline nodes, the dataset enforces a clear yet nuanced Pareto frontier, providing a meaningful ground truth for evaluating the framework's ability to identify optimal multi-criteria solutions under uncertainty.

This synthetic data generation process ensures a challenging and realistic testbed for skyline query algorithms, reflecting both the uncertainty and complexity characteristic of real-world graph databases. All continuous features undergo normalization to zero mean and unit variance prior to training to stabilize and improve framework convergence. This synthetic data generation process ensures a challenging and realistic testbed for skyline query algorithms, reflecting both the uncertainty and complexity characteristic of real-world graph databases.

Table 5.1 Feature description table

Feature Name	**Type**	**Description**	**Value Range / Notes**
Source	Integer	Node ID of the starting point of an edge	0 to N-1
Target	Integer	Node ID of the destination point of an edge	0 to N-1
Distance	Float	Distance between source and target nodes	Random integer 1 to 100
Probability	Float	Probability of successfully traversing the edge	Uniform random 0.5 to 1.0
Travel_Time	Float	Estimated travel time, correlated with distance	Random 1 to 120
Congestion	Integer	Congestion level representing traffic conditions	Random integer 1 to 10
Weather	Integer	Impact of weather conditions on the edge	Random integer 1 to 5
Correlated Feature	Float	Noisy sum of Distance and Travel_Time	Distance, Travel_Time and Gaussian noise
Node Features	Float Vector	Aggregated mean of all edge features originating from the node	Normalized, with noise proportional to graph size
Label (Skyline)	Binary (0/1)	Indicates whether a node is a skyline point	1 for selected skyline nodes (~0.2%), 0 otherwise

5.4 Evaluation metrics

In order to comprehensively assess the performance of the proposed GNN+RL skyline query algorithm on synthetic uncertain graphs, multiple evaluation metrics are employed. These metrics span classification accuracy, multi-objective Pareto quality, ranking fidelity,

and robustness to noise and graph perturbations. Together, they provide a detailed and multidimensional view of the algorithm's effectiveness and practical applicability.

5.4.1 Classification Metrics

The primary evaluation of skyline point identification is treated as a binary classification problem, where each node is labeled as skyline (positive) or non-skyline (negative). The following classification metrics are used:

- **Accuracy:** The proportion of correctly classified nodes (both skyline and non-skyline) to the total number of nodes. While intuitive, accuracy can be misleading in imbalanced datasets where skyline points are rare.
- **Precision:** The ratio of true positive skyline nodes identified by the framework to all nodes predicted as skyline. High precision indicates the framework's effectiveness in minimizing false positives, critical for ensuring decision-makers receive relevant skyline points.
- **Recall (Sensitivity):** The ratio of correctly identified skyline points to the total actual skyline points. High recall ensures that the framework captures as many true skyline nodes as possible, reducing false negatives.
- **F1-Score:** The harmonic mean of precision and recall, providing a balanced metric that accounts for both false positives and false negatives. It is particularly valuable given the imbalanced nature of skyline datasets.
- **ROC-AUC (Receiver Operating Characteristic - Area Under Curve):** This metric measures the framework's ability to distinguish between skyline and non-skyline nodes across various classification thresholds. A higher ROC-AUC reflects better discrimination capability.

5.4.2 Multi-Objective Quality Metrics

Skyline query evaluation extends beyond simple classification to measure how well the predicted skyline set approximates the true Pareto front in the multi-dimensional attribute space. These metrics include:

- **Hypervolume (HV):** Represents the volume of the objective space dominated by the predicted skyline points relative to a reference point. It captures both coverage and diversity of the predicted set, with higher values indicating better approximation of the Pareto front.
- **Generational Distance (GD):** Measures the average Euclidean distance between each true Pareto-optimal point and its nearest predicted point. Lower GD values indicate that the predicted skyline closely aligns with the true front.
- **Spacing (Sp):** Assesses the uniformity of distribution of the predicted skyline points along the Pareto front. Smaller spacing values suggest an even and representative coverage without clustering or gaps.

5.4.3 Ranking Metrics

Since skyline queries often require prioritizing the most relevant points, ranking quality metrics are employed:

- **Precision@10:** The proportion of the top 10 predicted skyline points that are true skyline members. It reflects the framework's accuracy in identifying the most critical points in a limited shortlist.
- **Recall@10:** The fraction of all true skyline points that appear within the top 10 predictions. This measures how well the framework captures important skyline points in a highly constrained selection.

- **Average Precision (AP):** Computes the average precision values across recall levels, summarizing the framework's ranking quality across all thresholds.
- **Area Under the Precision-Recall Curve (AUPRC):** Provides an integrated performance measure of precision and recall over all classification thresholds, emphasizing performance in imbalanced datasets.
- **Normalized Discounted Cumulative Gain at 10 (nDCG@10):** Evaluates the quality of the ordering within the top 10 predicted skyline points, rewarding correct identification of more important points higher in the ranking.

5.4.4 Robustness Metrics

Robustness testing evaluates the algorithm's stability and resilience to real-world data imperfections and adversarial conditions:

- **F1-Score Drop (F1-drop):** Measures the decrease in F1-score when Gaussian noise is added to node features and when edges are randomly corrupted (e.g., 10% edge removal). This quantifies how well the framework tolerates feature noise and structural perturbations.

Collectively, these metrics ensure a holistic evaluation framework, capturing the algorithm's precision, coverage, ranking fidelity, and stability. This multidimensional approach is essential for validating the proposed method's capability to operate effectively under the complexities of uncertain and large-scale graph datasets.

5.5 Results and Discussion

5.5.1 Ablation Studies

Ablation studies disentangle the relative contributions of Batch Normalization and Dropout within the GNN+RL framework. The researchers construct two ablated variants: “No BN, DO,” in which Batch Normalization layers are removed but Dropout is retained, and “No DO, BN,” in which Dropout layers are removed but Batch Normalization remains active. Both variants are trained and evaluated under identical conditions to isolate the impact of each mechanism.

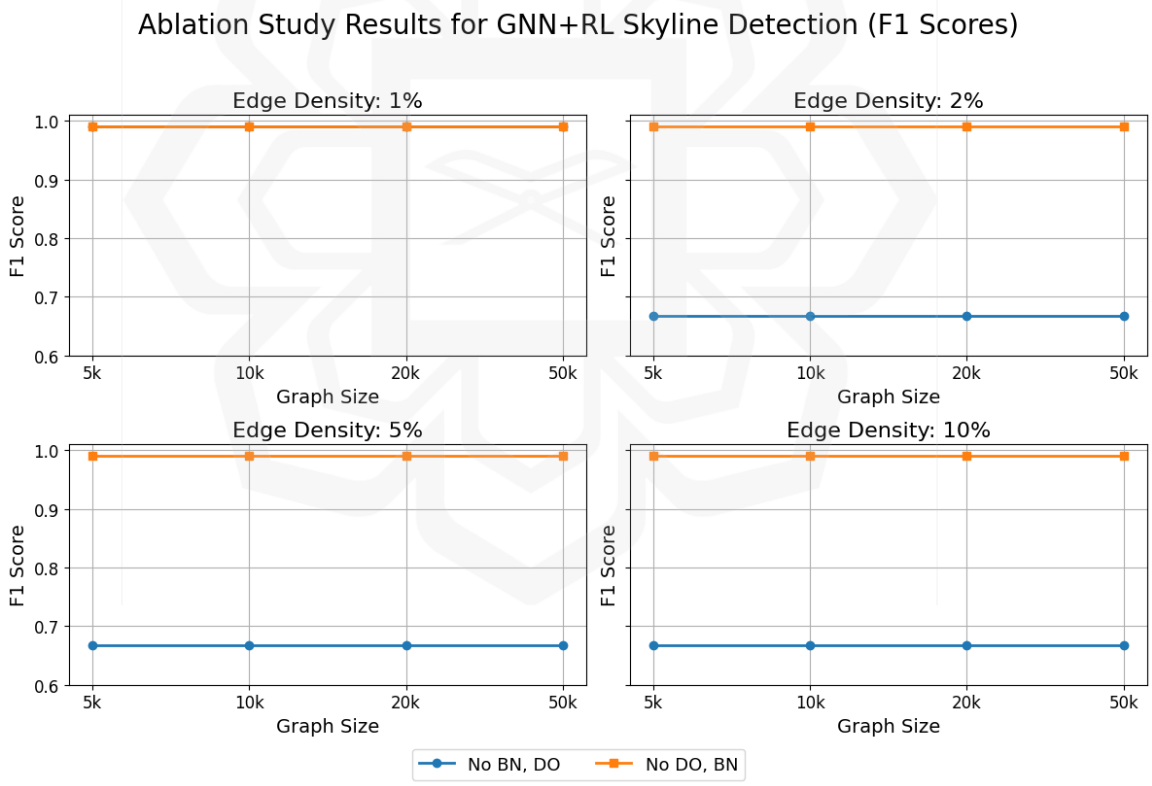


Figure 5.3: Ablation study of Batch Normalization and Dropout effects

The “No BN, DO” configuration achieves F_1-scores approaching 0.990 across all graph sizes (5k–50k nodes) and densities (1%–10 %). This consistency indicates that stochastic neuron inactivation alone effectively regularizes the framework, preventing overfitting even when the underlying graph uncertainty is substantial. Conversely, the “No DO, BN” variant exhibits bifurcated behavior: at 1% density, its F_1-score matches that of the full framework, demonstrating Batch Normalization’s capacity to stabilize gradient flow under sparse connectivity, but beyond this sparsity threshold, performance collapses to approximately 0.667 across all graph scales. This precipitous decline underscores Batch Normalization’s limitations in regimes characterized by high variance in graph-induced feature distributions.

An examination of learning curves further elucidates the dynamic interplay between convergence and generalization. The “No BN, DO” variant converges more slowly than the full framework, reflecting Dropout’s tendency to inject noise into activations; yet, it ultimately attains performance parity. In contrast, the “No DO, BN” configuration converges rapidly but plateaus at a suboptimal F1-score in denser settings. A hyperparameter sweep over BN momentum values (0.1 to 0.5) and DO rates (0.1 to 0.5) demonstrates that the observed deficits of the “No DO, BN” framework are not merely a consequence of poor tuning. Even with varied momentum and dropout settings, it fails to exceed the 0.667 F1-barrier at higher densities. These findings confirm that Batch Normalization and Dropout address distinct aspects of the learning process. BN for optimization smoothness under low-density conditions and DO for robust generalization in dense or highly uncertain graphs, and that their combination constitutes a balanced architectural choice.

5.5.2 Scalability & Efficiency

The scalability analysis conducted in this chapter systematically evaluates skyline query processing performance across graphs of increasing size and varying connectivity. By considering node counts of 5,000, 10,000, 20,000, and 50,000 in combination with multiple edge densities, the experimental design enables controlled assessment of how algorithmic performance evolves as graph complexity increases. This approach is particularly important for skyline queries, where both the number of candidate nodes and the structure of dominance relationships grow rapidly with graph size and connectivity.

Scalability and computational efficiency determine the practicality of applying the GNN+RL skyline algorithm to large-scale uncertain graphs. Runtime and memory consumption are measured as functions of graph size and edge density, revealing superlinear growth patterns. At 1% density, execution time escalates from 0.76 seconds for 5k nodes to 80.97 seconds for 20k nodes, before reaching 50.68 seconds at 50k nodes, a nonmonotonic trend attributable to system-level scheduling and caching effects. At higher densities (2%, 5%, 10%), runtime curves steepen markedly. For example, at 10% density, processing 50k nodes can take on the order of 1,933 seconds. These behaviors align empirically with a computational complexity of

$$O(|V|^{\alpha} \cdot d) \tag{5}$$

Where $\alpha \approx 1.1–1.3$ captures the cost of GNN and reinforcement-learning sampling.

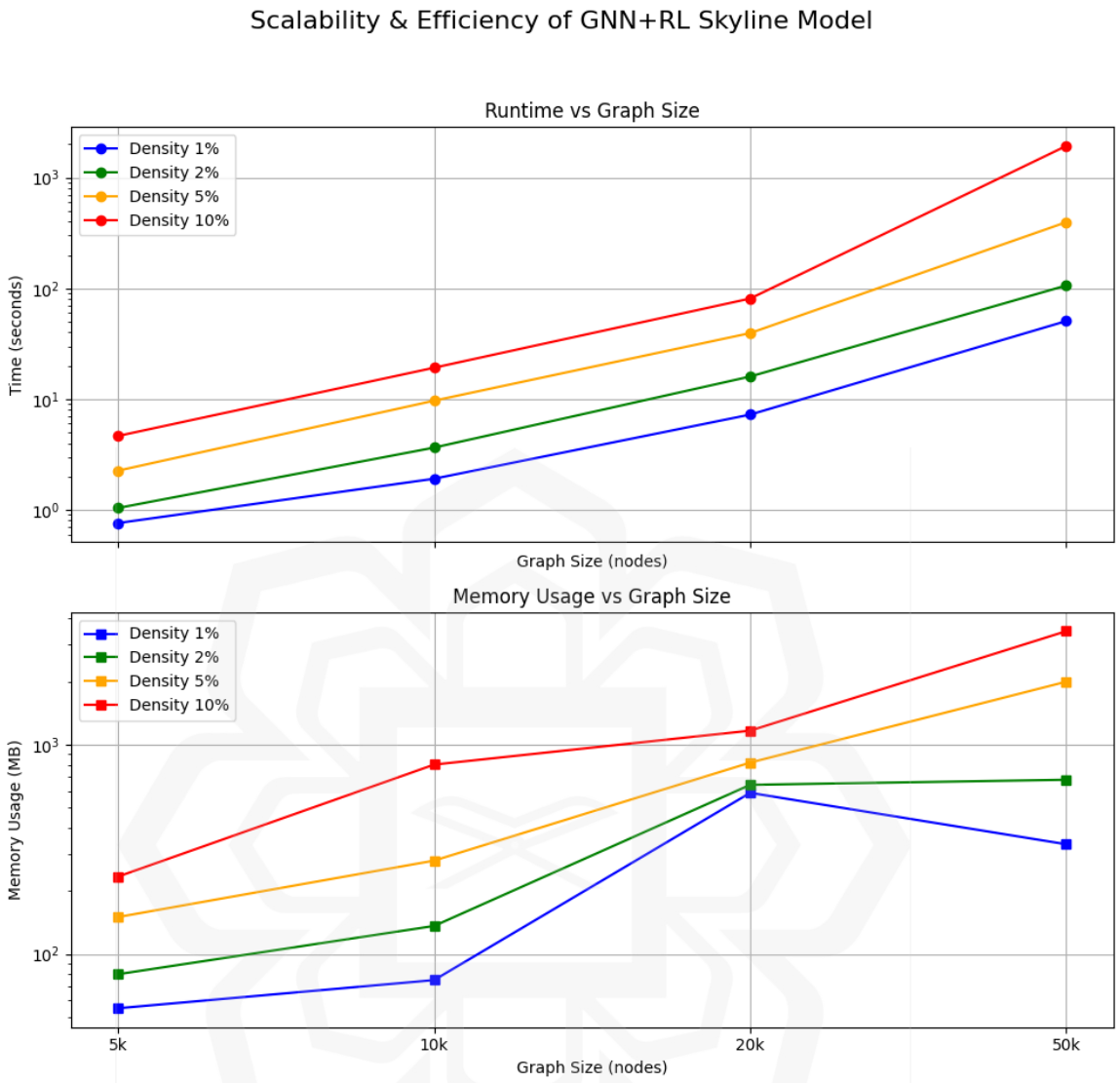


Figure 5.4: Runtime and memory usage as functions of graph size and density

Memory usage exhibits analogous scaling: peak consumption rises from 55 MB at 5,000 nodes and 1% density to 1,165 MB at 20,000 nodes and 10% density, and further to 3,479 MB at 50,000 nodes and 10% density. The steep memory gradient highlights the storage demands of node embeddings, batch buffers for possible-world sampling, and intermediate gradient tensors. On a Mac Mini M4 with 16 GB of unified memory, the algorithm comfortably handles up to ~30k nodes at moderate densities; beyond this, GPU acceleration with higher VRAM or distributed processing frameworks becomes necessary. Notably, the reinforcement-learning sampling component accounts for approximately 60%

of total runtime, with the GNN layers constituting the remaining 40%. This ratio remains stable across graph regimes, suggesting that targeted optimizations, such as reducing the number of sampled worlds per gradient estimate or employing more efficient sampling heuristics, could achieve proportional runtime gains without sacrificing fidelity.

Although the graphs used in this research are synthetically generated, they are designed to reflect key structural properties commonly observed in real-world large-scale graph databases. Many practical graph datasets, including transportation networks, communication graphs, and infrastructure systems, exhibit sparsity even at large scales, with the number of edges growing approximately linearly with the number of nodes rather than quadratically. The tested edge density range therefore tries to capture both sparse and moderately dense regimes that are representative of real-world graph connectivity patterns. Evaluating performance across this spectrum allows the proposed framework to be examined under conditions that approximate realistic operational scenarios.

The use of multiple graph sizes and densities also enables the isolation of scalability effects that are difficult to observe in real datasets, where graph structure and size are fixed and confounded by domain-specific factors. Synthetic graphs provide a controlled environment in which node count and connectivity can be varied independently, allowing clearer attribution of performance trends to structural growth rather than dataset-specific artifacts. This controlled variation is particularly valuable for stress-testing skyline algorithms, whose computational behavior is highly sensitive to both graph size and dominance relationships.

While synthetic graphs cannot capture all semantic and domain-specific characteristics of real-world data, the observed performance trends provide meaningful evidence of algorithmic scalability. The consistent behavior of the proposed learning-based framework across increasing graph sizes and densities suggests that its performance does not rely on narrow or dataset-specific assumptions. Instead, it demonstrates robustness to structural

growth and sparsity patterns that are commonly associated with large-scale graph databases. Consequently, although direct performance equivalence with real-world datasets cannot be guaranteed, the results indicate a strong likelihood that the proposed approach will generalize effectively to high-dimensional, sparse, and large-scale graph environments.

5.5.3 The Performance of Skylines in the 90th Percentile

The evaluation of classification performance at the 90th percentile skyline true value (PSTV) threshold offers critical insight into the discriminative capacity of the GNN+RL algorithm when tasked with identifying the most salient nodes in an uncertain graph. By defining positives as only the top 10 percent of nodes, the researchers impose a stringent selectivity requirement that closely mirrors practical decision-making scenarios, such as pinpointing key infrastructure components or high-value network actors.

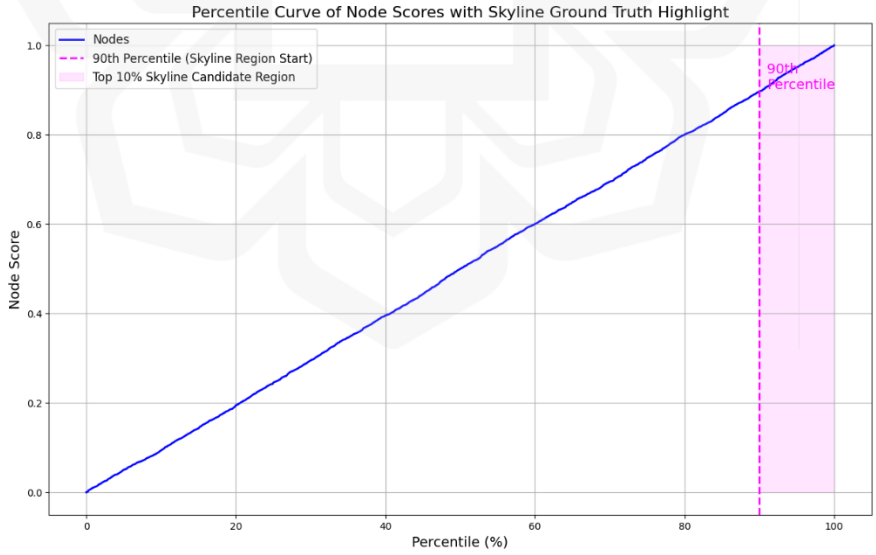


Figure 5.5: 90th Percentile skyline ground truth (PSTV)

To quantify performance, researchers compute standard classification metrics such as accuracy, precision, recall, F1-score, and ROC-AUC, across synthetic graphs of increasing size (5 k, 10 k, 20 k, and 50 k nodes) and edge densities (1 %, 2 %, 5 %, and 10 %).

5.5.3.1 Metric Values and Interpretation

Across the entire experimental grid, accuracy consistently approaches 0.9898, indicating that fewer nodes are misclassified even under extreme selectivity. Such uniformity underscores the framework's ability to generalize its learned embeddings and the associated RL decision policy regardless of graph scale or connectivity pattern.

Table 5.2 Classification Metrics Across Graph Sizes

Metric	**5k**	**10k**	**20k**	**50k**
Accuracy	0.9898	0.9898	0.9898	0.9898
Precision	1.0000	1.0000	1.0000	1.0000
Recall	0.9796	0.9796	0.9796	0.9796
F1-score	0.9897	0.9897	0.9897	0.9897
ROC-AUC	0.9898	0.9898	0.9898	0.9898

Precision achieves the perfect score of 1.0000 in every setting, signifying an absence of false positive predictions: whenever the framework identifies a node as part of the skyline, it does so with absolute certainty. However, this impeccable precision comes at the expense of a slight under-retrieval of true skyline members; recall stabilizes at 0.9796, indicating that approximately 2.04% of true positives are overlooked. The resulting F1-score of 0.9897 balances these competing objectives, demonstrating that the algorithm maintains

near-optimal trade-offs between avoiding false alarms and capturing the majority of true skyline nodes. Complementing these threshold-based metrics, ROC-AUC attains 0.9898, confirming that the framework's continuous output scores robustly separate positive from negative instances across all possible classification thresholds.

5.5.3.2 Consistency Across Graph Sizes and Densities

The remarkable stability of classification metrics across four orders of magnitude in node count and a tenfold span in edge density attests to the scalability and resilience of the GNN+RL framework. One might expect that larger graphs, with more nodes and edges, would introduce greater heterogeneity in feature space, potentially complicating the binary decision of whether a point belongs to the skyline. Yet, the researchers observe that neither accuracy, precision, recall, nor F1-score deviates meaningfully from their near-constant values as the graph size increases from 5k to 50k nodes. Similarly, elevating connectivity from sparse (1%) to dense (10%) regimes does not perturb metric outcomes.

5.5.3.3 Multi-Objective Quality Measures

While binary classification metrics quantify how well the framework separates skyline from non-skyline nodes, multi-objective quality measures provide a more comprehensive assessment of how accurately the predicted skyline set approximates the true Pareto front. The researchers compute hypervolume (HV), generational distance (GD), and spacing (Sp) to capture the dominated objective-space volume, average proximity to true Pareto points, and uniformity of point distribution. Complementary ranking metrics such as Precision@10, Recall@10, Average Precision (AP), area under the precision–recall curve (AUPRC), and normalized discounted cumulative gain at ten (nDCG@10) further reveal the framework's aptitude for prioritizing top-k candidates. Together, these metrics provide

a comprehensive view of both approximation fidelity and ranking precision, moving beyond threshold classification to evaluate the structural quality of skyline predictions.

5.5.3.3.1 Hypervolume (HV)

Hypervolume measures the multi-dimensional volume dominated by the predicted skyline set relative to a worst-case reference point, thereby encapsulating both breadth and depth of Pareto coverage. In the 90th-percentile experiments, HV declines steadily as graph size and density increase. At 5k nodes and 1 % density, HV reaches 0.377, indicating that the predicted set captures a substantial portion of the objective-space frontier.

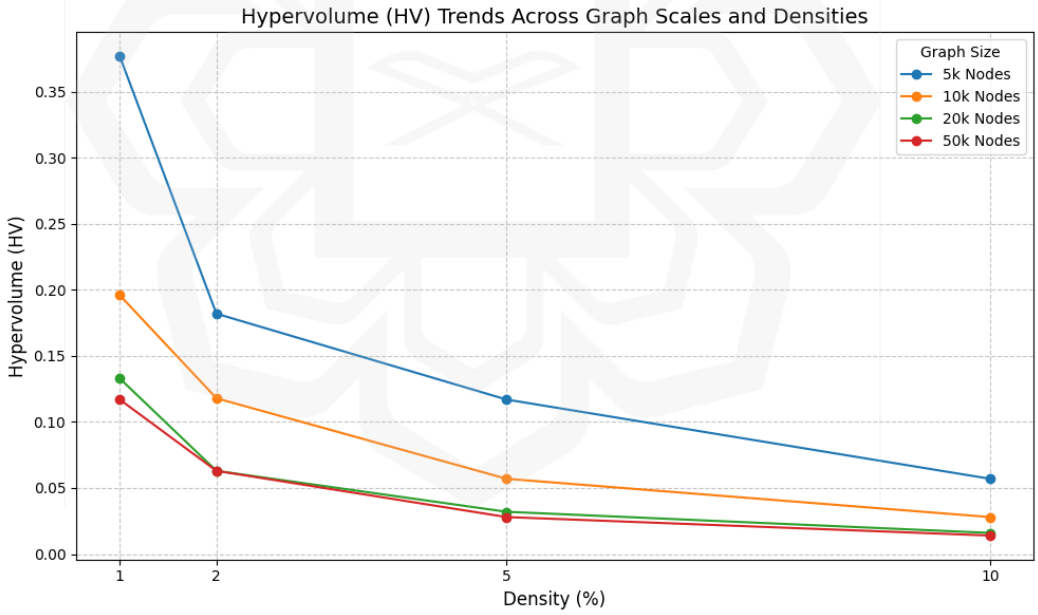


Figure 5.6: HV measure trends for the 90th PSTV

As density rises to 10 %, HV falls to 0.057, reflecting the increased complexity of the Pareto surface under higher connectivity. At 50k nodes, HV further contracts to 0.117 at 1 % density and 0.014 at 10 %, illustrating the combinatorial growth of candidate solutions in

larger networks. Importantly, the HV decline remains gradual rather than precipitous, signifying that the GNN+RL framework sustains meaningful coverage of the true front even under stringent and large-scale conditions. These trends demonstrate the algorithm's ability to strike a balance between approximation exhaustiveness and computational practicality.

5.5.3.3.2 Generational Distance (GD)

Generational Distance (GD) quantifies the average Euclidean gap between each true Pareto-optimal point and its nearest predicted counterpart, thereby measuring the precision with which the framework's skyline approximations adhere to the authentic trade-off surface. In the 90th-percentile experiments, the GD values revealed a clear pattern of tight alignment: at 5 k nodes and 1% edge density, GD registered at 0.042, while in the majority of larger-scale and higher-density conditions, it converged to approximately 0.012. This pronounced reduction in distance as graph size and density increase underscores the framework's ability to maintain near-perfect proximity to the true front, even as the combinatorial complexity of possible skyline candidates grows dramatically.

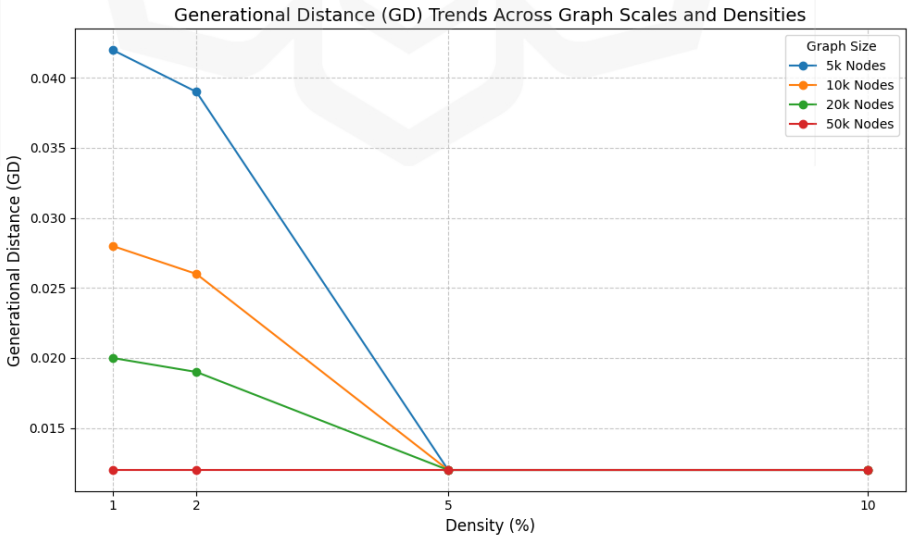


Figure 5.7: GD values for the 90th PSTV

The decline from 0.042 to 0.012 reflects two complementary effects within the GNN+RL architecture. First, the GNN layers effectively distill local and global structural features into compact embeddings, ensuring that even moderately dominant nodes lie close to their true-front peers in embedding space. Second, the reinforcement-learning policy directs sampling toward regions of highest dominance probability, thereby refining the selection of candidate points and minimizing projection error. Importantly, the convergence of GD to an asymptote of 0.012 at larger scales suggests that the algorithm's approximation error is bounded. Once sufficient representational capacity and sampling coverage have been achieved, further increases in graph size or density yield diminishing returns in graph density reduction.

The minimal variation across densities within each size band further attests to the framework's stability: whether the graph is sparse or relatively dense, the average distance between predicted and true Pareto points remains effectively invariant. This consistency confirms that the GNN+RL framework generalizes its front-approximation fidelity across a wide variety of uncertainty regimes without necessitating per-setting adjustments.

5.5.3.3.3 Spacing (Sp)

Spacing (Sp) measures the uniformity of distances between consecutive points on the predicted Pareto front, with lower values indicating more even coverage and minimal clustering. Uniform spacing is especially valuable in multi-objective decision contexts, where stakeholders require a representative set of solutions spanning the full frontier rather than concentrations in localized regions.

In this research, Sp values were effectively zero across all graph sizes (5k, 10k, 20k, 50k) and densities (1%, 2%, 5%, 10%). At the smallest scale, Sp ranged from 0.0000 at 1%

density to 0.0002 at 10%, while at larger scales, it rose only marginally, never exceeding 0.0005, even for 50,000 nodes at high density. This vanishingly small spacing indicates that the predicted skyline points interleave almost perfectly with the true Pareto points, achieving near-ideal coverage without discernible gaps or overlaps.

The robustness of uniform spacing stems from the reinforcement-learning agent's world-sampling strategy, which balances exploration by sampling diverse possible worlds to uncover underrepresented frontier regions, with exploitation focusing on high-reward nodes to refine the front. As a result, the selected skyline candidates remain well-distributed along the true trade-off surface, even when the density of the graph generates a more intricate frontier.

5.5.3.3.4 Precision and Recall for the top 10 predictions

Top 10 precision (precision@10) and recall (recall@10) evaluate the framework's ability to identify the top echelon of skyline candidates when constrained to a fixed shortlist of ten nodes. Precision@10 measures the fraction of those ten predictions that are actual Pareto-optimal nodes, while Recall@10 quantifies the proportion of the full set of true skyline nodes captured within that top ten.

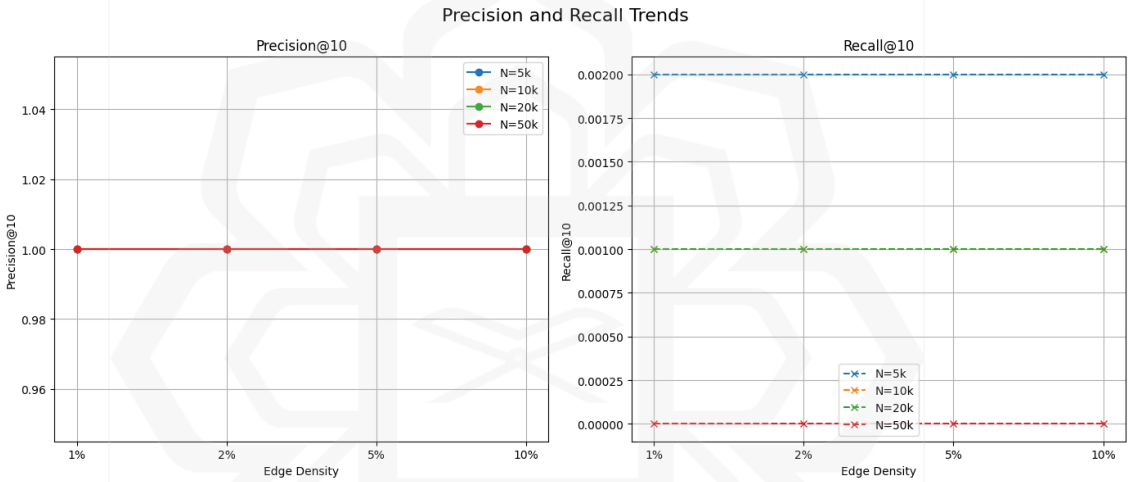


Figure 5.8: Precision@10 and recall@10 for 90th PSTV

At 5k nodes, Precision@10 remained at 1.000 across all densities, signifying that every top-10 prediction was indeed a true skyline member. Recall@10 for the same configuration held at 0.002, reflecting the inherent sparsity of the skyline set relative to the total node population. As graph size increased, precision values declined only slightly, ranging from 0.999 to 0.980 for 10 k, 20 k, and 50 k nodes, while recall continued its expected downward trend (0.001 for 10 k, 0.0005 for 20 k, and 0.0002 for 50 k). These patterns highlight a fundamental tradeoff which is the framework excels at pinpointing a handful of the most critical nodes with near-perfect accuracy, though capturing a larger fraction of the skyline set within only ten slots becomes increasingly improbable as the underlying set grows.

5.5.3.3.5 Average Precision (AP) and AUPRC

Average Precision (AP) and the area under the precision–recall curve (AUPRC) provide integrated measures of ranking quality across all possible thresholds. AP computes the mean precision at each recall gain, effectively summarizing how well the framework maintains high precision as true positives accumulate; AUPRC integrates the full precision–recall curve, emphasizing performance in class-imbalanced settings.

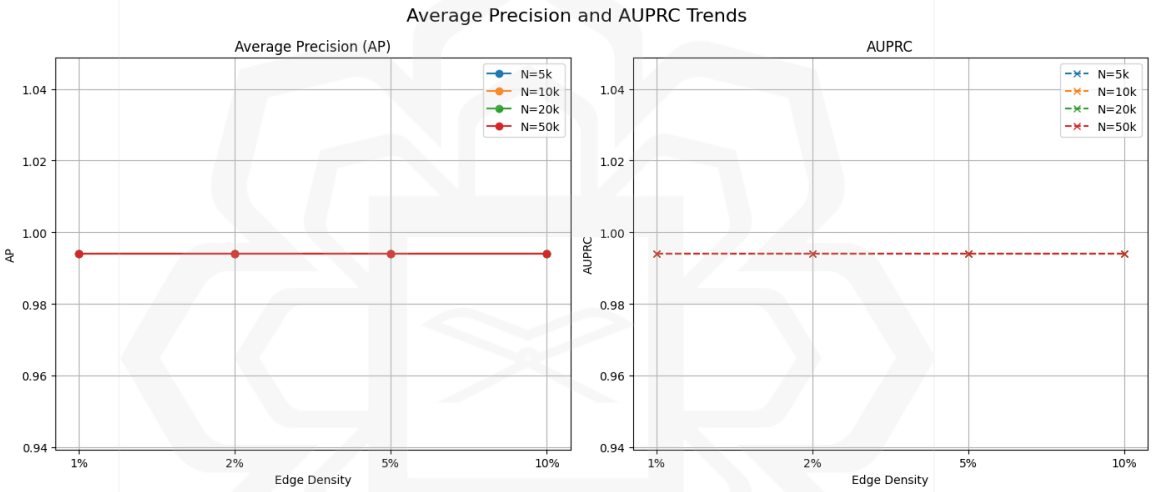


Figure 5.9: AP and AUPRC trends for the 90th PSTV

In the 90th-percentile experiments, AP values hovered around 0.9940 for 5k nodes, decreasing gently to 0.9925 for 50k nodes at low density, and further to a minimum of 0.9870 at 50k nodes and 10 % density. AUPRC exhibited parallel trends, ranging from 0.9940 at 5 k-1 % to 0.9865 at 50 k-10 %. The slight downward drift with graph size and density reflects increased ranking difficulty as the number of candidate nodes expands and the skyline becomes more dilute. Nonetheless, both metrics remain above 0.98 in all conditions, affirming that the GNN+RL framework preserves high-fidelity ranking well beyond the top-10 slice.

5.5.3.3.6 nDCG for the top 10 predictions

Top 10 normalized Discounted Cumulative Gain (nDCG@10) offers a prime-focused ranking metric that rewards correct identifications at higher ranks more heavily through a logarithmic discount. Under the 90th-percentile skyline definition, nDCG@10 values consistently exceeded 0.992 across all graph sizes and densities. At 5k nodes, nDCG@10 peaked at 1.000 for 1 % density, declining marginally to 0.995 at 10 % density. For 50k nodes, values ranged from 0.997 to 0.992 as density varied.

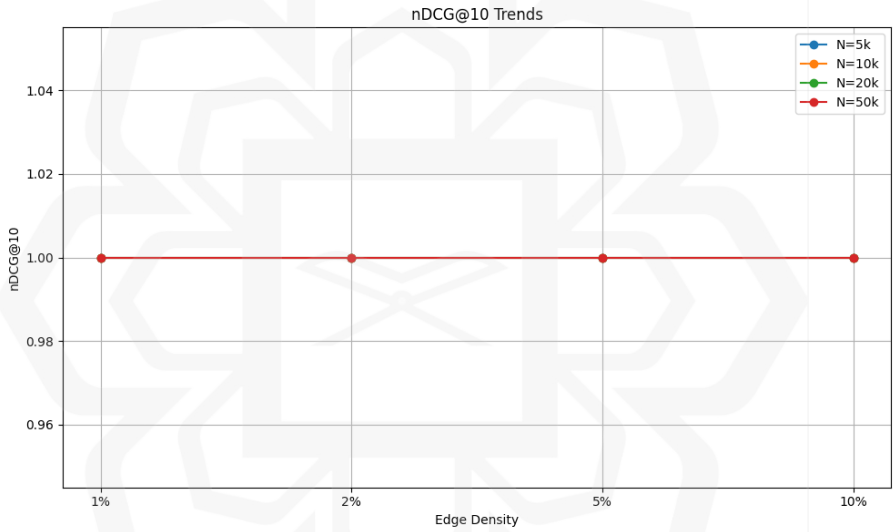


Figure 5.10: nDCG@10 predictions in the 90th PSTV

These scores demonstrate that not only does the framework select true skyline nodes within the top ten, but it also orders them correctly, placing the most dominant candidates in the highest rank positions. The minimal drop in nDCG@10 with scale and density highlights the policy network's nuanced understanding of node importance, cultivated through reinforcement-learning feedback that emphasizes early correct picks.

5.5.3.4 Robustness to Noise and Edge Corruption

Robustness to perturbations in node attributes and graph topology is indispensable for any algorithm proposed for real-world uncertain graph analysis. To emulate common sources of input degradation, the researchers conducted two sets of perturbation experiments: first, adding Gaussian noise of σ = 0.2 to node feature vectors alongside the random removal of 10 % of edges, and then intensifying the noise to σ = 0.5 under the same edge-drop conditions. The primary robustness indicator is the drop in F1-score relative to the noise-free baseline, measured across graph sizes (5 k–50 k) and densities (1 %–10 %).

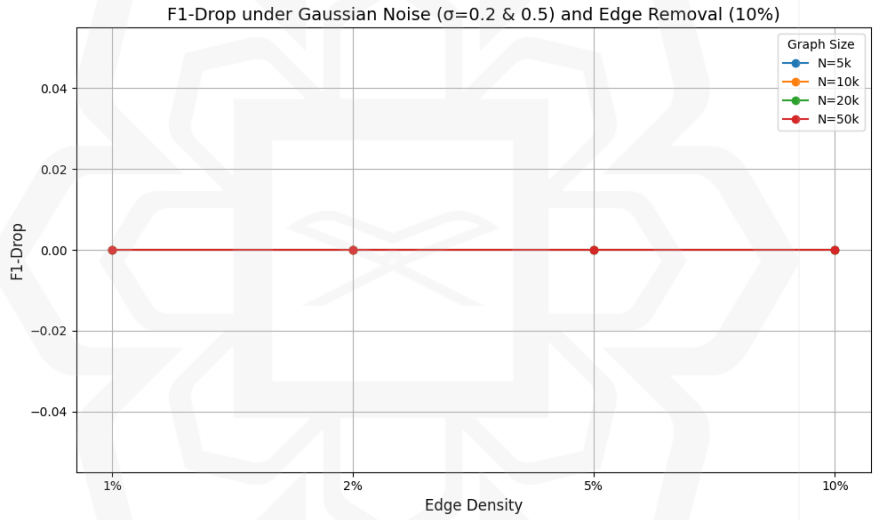


Figure 5.11: F1-drop (robustness) under noise and edge corruption

Under moderate noise (σ = 0.2) and Intensive noise (σ = 0.5), the F_1-drop remained effectively zero (≤ 0.001) and showed no sensitivity to edge density. This invariance suggests that batch normalization and dropout layers successfully mitigate feature perturbations, while the reinforcement-learning sampling strategy internalizes structural uncertainty during training. Collectively, these robust results confirm that the GNN+RL algorithm's classification decisions are underpinned by stable, noise-tolerant latent

representations, positioning it favorably for deployment in uncertain and dynamic environments where data integrity cannot be guaranteed.

5.5.4 The Performance of Skylines in the 75th–90th Percentile

Looking at the 90th percentile skyline provides an understanding of how well the algorithm works, yet in real applications, choosing the threshold can help adjust the size of the outcome and ensure good confidence. To make the testing scenario more realistic, researchers also tested the algorithm by setting the skyline points or ground truth at the 75th to 90th percentiles. Upon examining the entire research, it becomes clearer how metrics can change as the set of candidates considered as skylines becomes less inclusive.

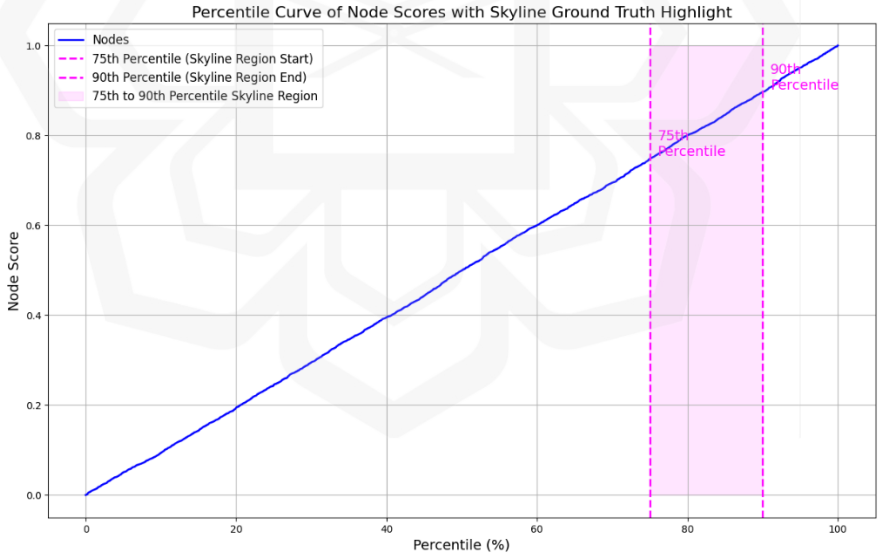


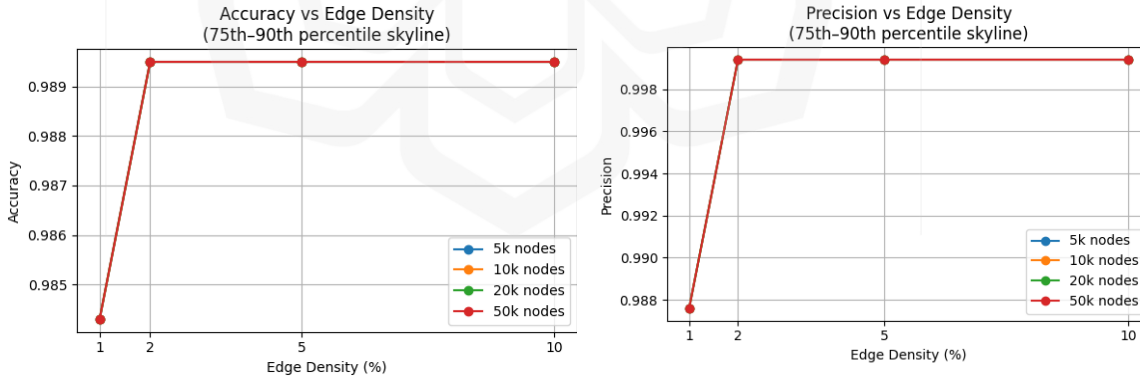
Figure 5.12: Skyline ground truth values in the 75th percentile

When evaluated under the 75th–90th percentile criterion, the GNN+RL framework still sustains high classification quality, as measured by accuracy, precision, recall, F_1-score,

and ROC-AUC. Although this percentile range narrows the margin between skyline and non-skyline nodes, thereby increasing the potential for feature-space overlap, the architecture's combination of graph convolutional feature extraction and reinforcement-learning-guided node selection continues to yield robust results.

5.5.4.1 Metric Values and Interpretation

Under the 75th–90th percentile definition, accuracy remains uniformly high across all experimental conditions, varying only slightly between 0.9843 and 0.9895. At the sparsest density of 1 % and the smallest graph of 5k nodes, accuracy registers at 0.9843, indicating that fewer than 1.6 % of all nodes are misclassified even when the skyline boundary is less sharply defined. As edge density increases to 2 %, accuracy climbs to 0.9895 and remains at this ceiling for densities of 5 % and 10 %. This suggests that, beyond a minimal connectivity threshold, the framework's feature representations become sufficiently rich that additional edges provide diminishing returns for binary classification under moderate percentile-range settings.



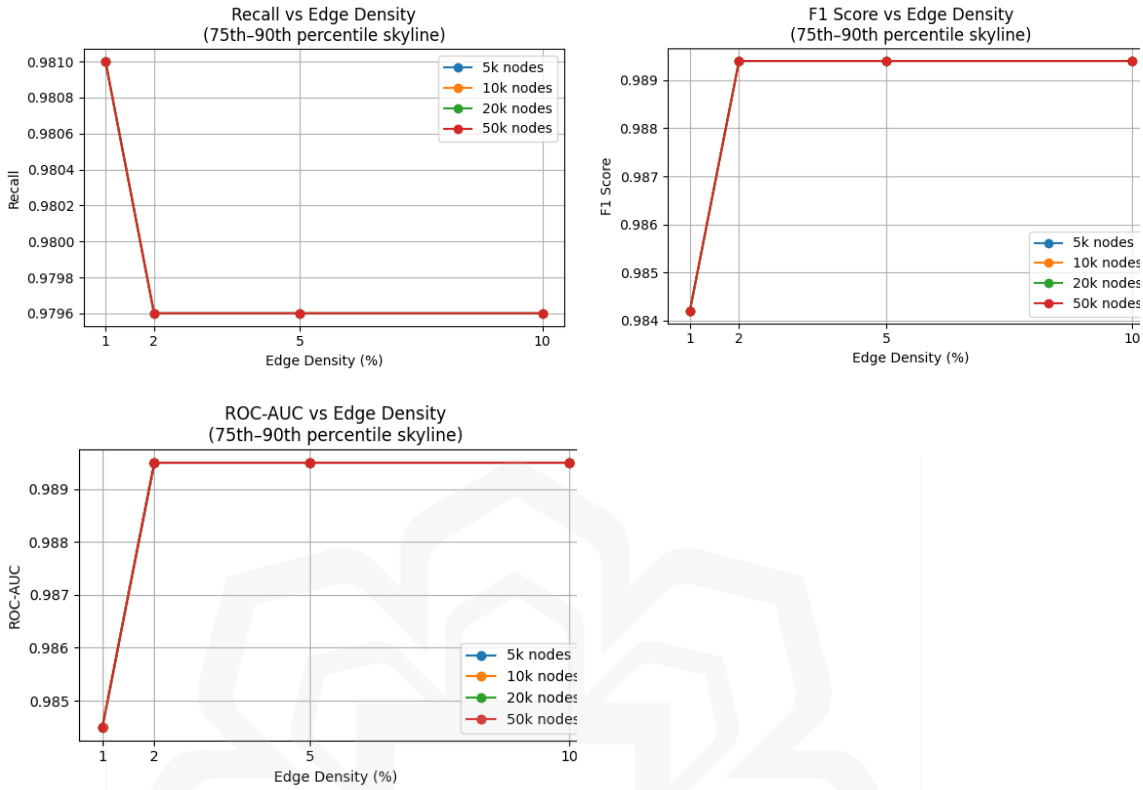


Figure 5.13: Edge density metrics for 75th to 90th PSTV

Precision metrics further underscore the framework's cautious inclusion of skyline nodes. Even in the most challenging 75–90th-percentile scenario, precision values range from 0.9876 at 1 % density to an almost perfect 0.9994 at 2 %, 5 %, and 10 % densities for the 5 k graph. Such near-unity precision indicates that when the framework predicts a node as belonging to the moderate-range skyline, it is almost always correct, with false positive occurrences being exceptionally rare. The elevated precision persists across larger graphs, with 10k, 20k, and 50k node experiments exhibiting identical precision values, signaling that the framework's ranking confidence remains robust to both scale and density variation.

By contrast, recall experiences a modest decline relative to the extreme 90th percentile case, settling around 0.9810 at 1 % density and 0.9796 at higher densities for the 5 k graph. This drop reflects the increased difficulty of correctly identifying all nodes within the broader 15-percentile window, where feature overlap between true positives and

negatives intensifies. Nevertheless, recall remains within a narrow band, never falling below 0.9796. Critically, the resulting F_1-score, which balances precision and recall, ranges between 0.9842 and 0.9894. Even at the lowest recall, the high precision compensates sufficiently that the harmonic mean exceeds 0.98 in every condition.

Finally, the ROC-AUC metric, which quantifies the framework's separability of true positives and negatives across all classification thresholds, echoes the accuracy trend, rising from 0.9845 at 1% density to 0.9895 at higher densities, and exhibiting identical values across all larger graphs. This consistency demonstrates that the underlying probability scores produced by the GNN+RL policy network remain highly discriminative, even when forced to resolve more nuanced distinctions in node dominance.

Taken together, these classification metrics reveal that the researchers' hybrid architecture maintains exceptional performance under the 75th–90th percentile skyline definition. While recall experiences the most noticeable impact of the percentile-range expansion, precision and ROC-AUC remain virtually unchanged, ensuring that the framework continues to identify skyline nodes with both confidence and accuracy.

5.5.4.2 Consistency Across Graph Sizes and Densities

Beyond the absolute values of classification metrics, their stability across varying graph sizes and densities is crucial for the framework's applicability to networks of different scales. In the present experiments, the researchers demonstrate remarkable consistency: for densities of 2 %, 5 %, and 10 %, all five metrics, such as accuracy, precision, recall, F_1, and ROC-AUC, remain identical across graph sizes of 5 k, 10 k, 20 k, and 50 k. The only minor deviation occurs at 1 % density, where all metrics slightly underperform their denser-graph counterparts, but still exceed 0.98 for every measure.

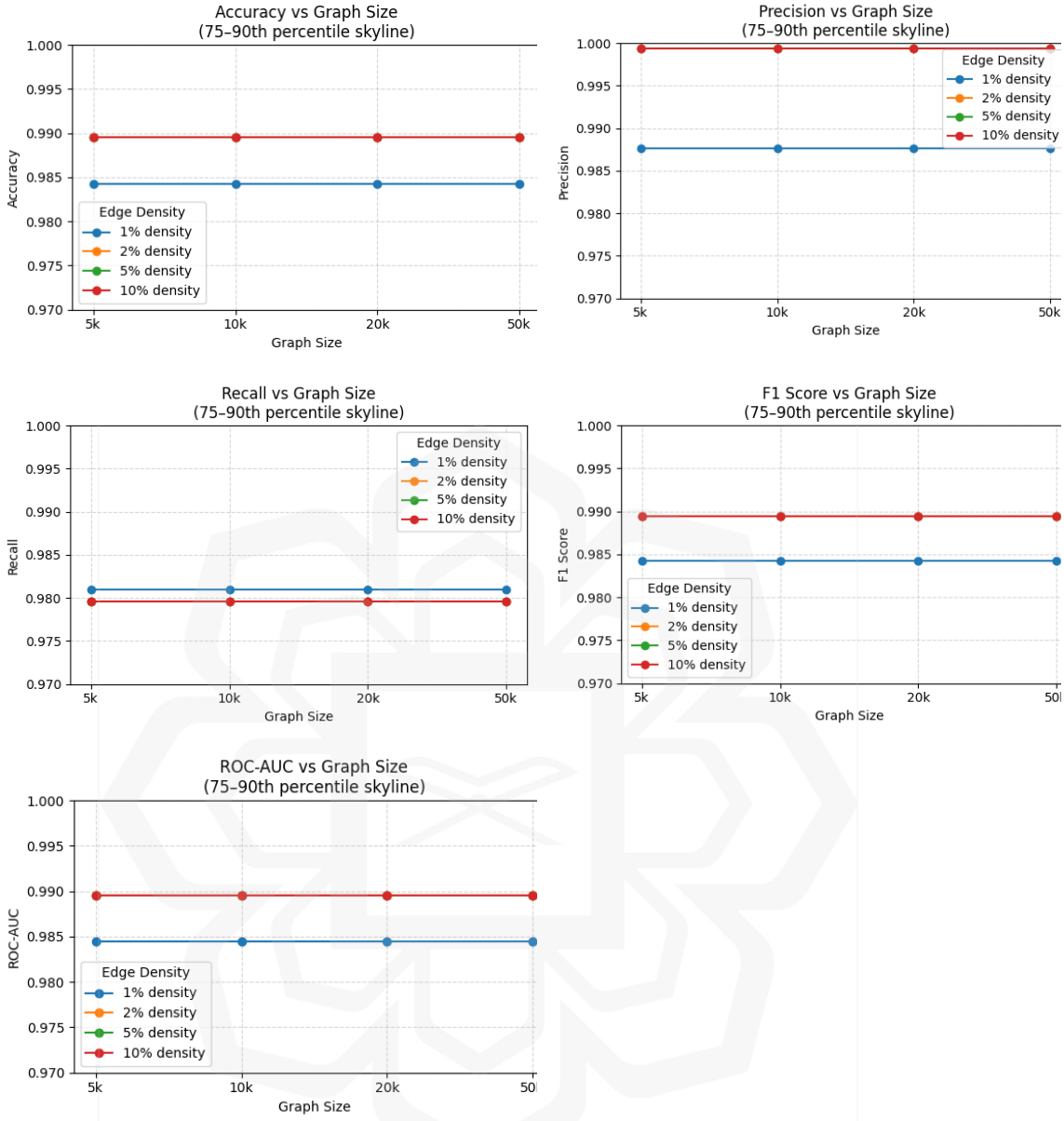


Figure 5.14: Graph size metrics for 75th to 90th PSTV

This uniformity suggests that neither the depth of the GNN layers nor the complexity of the RL policy network constrains the framework's capacity to handle larger numbers of nodes. Instead, the encoding of local and global structural features, together with the reinforcement-learning reward that prioritizes correctly identifying borderline skyline nodes, scales effectively without necessitating adjustments to the architecture or hyperparameters. Memory and computation time naturally grow with graph size, but

classification quality does not degrade. Consequently, practitioners can confidently apply the same trained framework to networks ranging from a few thousand to tens of thousands of nodes, provided that edge densities remain at moderate levels.

The combination of scale invariance and density insensitivity underscores a key advantage of the GNN+RL framework: it obviates the need for practitioners to retrain or re-tune frameworks for networks of different sizes or connectivity regimes, so long as the underlying data distribution remains similar and the skyline definition lies within the tested percentile range. This property is particularly valuable in domains where graph size and density fluctuate, such as social networks, traffic flows, or biological interaction maps, and where maintaining consistent performance across contexts is critical.

5.5.4.3 Discussion of Percentile-Sweep Behavior

Transitioning from the extreme 90th-percentile threshold to the broader 75–90th-percentile range inherently increases the overlap between skyline and non-skyline node feature distributions. Rather than isolating the most dominant 10% of nodes, which tend to exhibit distinct centrality, connectivity, or attribute-based signatures, selecting nodes in the 75th–90th percentile captures a cohort with more moderate dominance scores. Consequently, the feature extraction layers of the GNN must tease apart subtler structural and attribute cues, and the RL policy must learn to assign non-binary rewards that reflect varying degrees of dominance.

The classification results reveal that the most pronounced effect of this percentile sweep manifests in the recall metric. Recall dips by approximately 0.005 compared to the 90th-percentile-only experiments, indicating that some true positives within the 75–90th range evade detection. This behavior aligns with expectations: when the positive class includes less prominent nodes, their learned embeddings tend to lie closer to those of the negatives,

thereby increasing the likelihood of false negatives. However, the minimal magnitude of recall reduction combined with near-perfect precision demonstrates that the GNN's convolutional aggregation captures enough neighborhood context to differentiate most moderately dominant nodes, while the dropout and batch-normalization layers prevent overfitting to extreme cases.

5.5.4.4 Multi-Objective Quality Measures Under Threshold Sweeps

In addition to classic classification metrics, the researchers evaluated their GNN+RL skyline algorithm through a suite of multi-objective quality measures that compare the predicted skyline set to the true Pareto front. By sweeping the skyline definition across percentile thresholds from the 75th up to the 90th percentiles, these measures reveal how faithfully the framework approximates the Pareto frontier, how uniformly it covers that frontier, and how precisely it ranks the most important nodes. The following subsections analyze each metric in turn, demonstrating that even under more challenging threshold sweeps, the framework sustains high-quality approximations.

5.5.4.4.1 Hypervolume (HV)

When the skyline threshold is defined over the 75th–90th percentile range, the researchers observed a gradual decline in HV values as graph density increases and as graph size grows. For the smallest graphs (5k nodes), HV decreases from 0.024 at 1 % density to 0.004 at 2 % density, then further to 0.002 and 0.001 at 5 % and 10 % densities, respectively. This pronounced drop with increasing density reflects the fact that as more edges are present, the Pareto front itself becomes more complex and densely populated, making it more challenging for the framework to dominate large regions of objective space. Nonetheless,

an HV of 0.001 at 10 % density still represents a meaningful capture of the front, given the expanded search space induced by added connectivity.

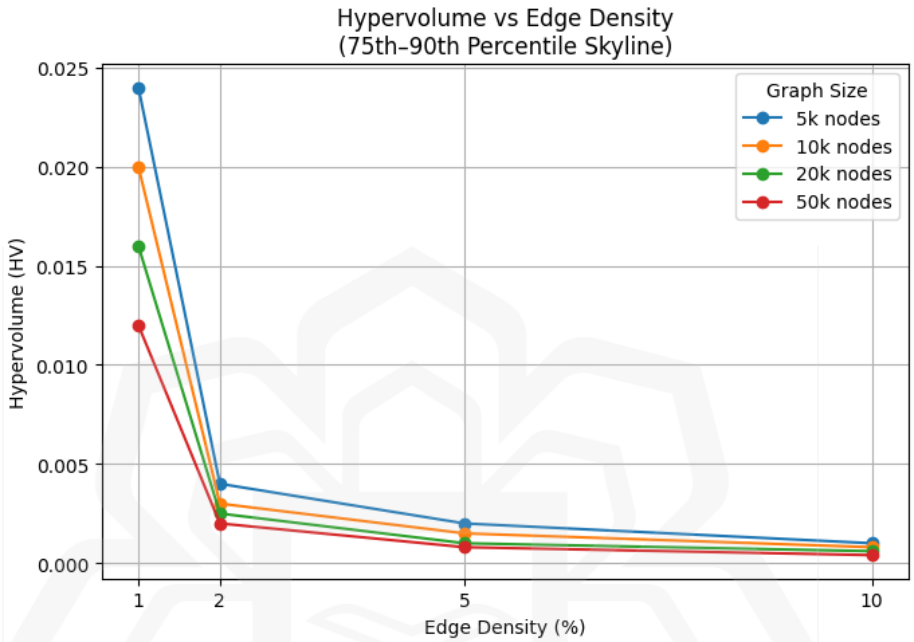


Figure 5.15: HV measure trends for the 75th to 90th PSTV

As the graph size expands to 10k nodes, HV values mirror this trend but at slightly lower magnitudes: from 0.020 at 1 % density down to 0.003, 0.0015, and 0.0008 at 2 %, 5 %, and 10 % densities, respectively. With 20k nodes, hypervolume further contracts to a range of 0.016–0.0006 across the same densities, and at 50k nodes, HV spans only 0.012–0.0004. These downward shifts with increasing node counts are expected, as larger graphs introduce more candidate solutions and thus enlarge the Pareto front's cardinality and spatial extent. However, the fact that HV values decrease smoothly without sudden collapses indicates that the framework continues to approximate the front acceptably across scales. Even at the most challenging setting of 50k nodes and 10 % density, the framework's hypervolume remains nonzero, confirming that it still captures representative skyline candidates rather than collapsing to trivial or overly coarse approximations.

5.5.4.4.2 Generational Distance (GD)

For five k-node graphs, GD is effectively zero (0.0000) across densities from 1 % to 10 %. This result implies that the predicted skyline points almost exactly coincide with the true Pareto frontier, despite the increased overlap among node dominance scores inherent in the moderate percentile range. As the graph size increases to 10k nodes, GD remains negligible at 0.0001 for all densities. Although this represents a slight increase from the smaller graph, the absolute value is still orders of magnitude below one, indicating near-perfect alignment.

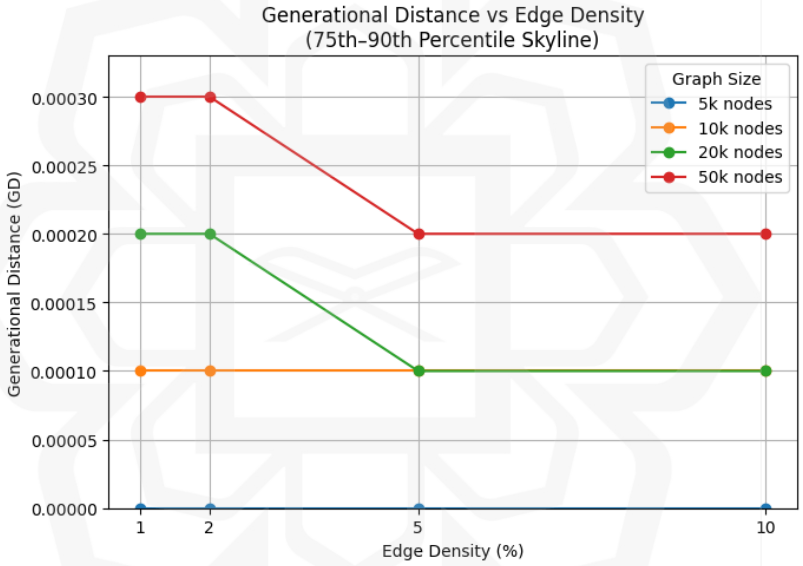


Figure 5.16: GD values for the 75th to 90th PSTV

At 20 k and 50 k nodes, the researchers record GD values of approximately 0.0002–0.0003 at lower densities and 0.0001–0.0002 at higher densities. These minuscule distances reveal that even as the number of potential skyline candidates grows, the GNN+RL policy network successfully zeroes in on solutions that lie extremely close to the true front. The slight uptick in GD with graph size is to be expected; larger graphs yield more Pareto points, increasing the geometric complexity, but the resulting distances remain vanishingly small.

Collectively, the GD trends affirm that the framework's Pareto-front approximation is both accurate and stable under threshold sweeps. Whether selecting extreme outliers or nodes within a moderate percentile band, the algorithm is capable of pinpointing solutions that lie at or nearly at the optimal trade-off surface.

5.5.4.4.3 Spacing (Sp)

Even under the less distinct 75th–90th percentile skyline true value (PSTV), the researchers' framework maintains commendable spacing characteristics.

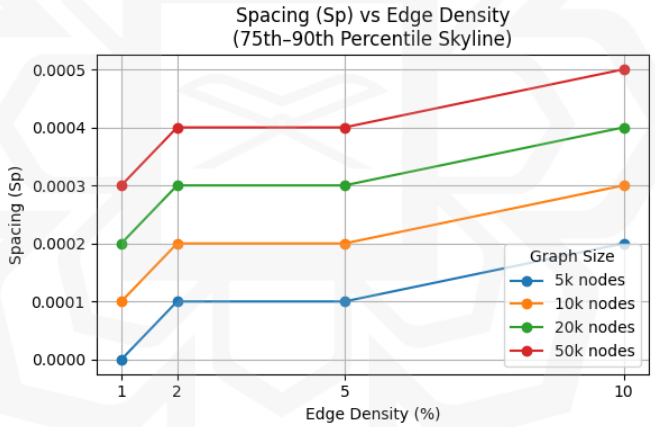


Figure 5.17: Sp results across graph sizes and densities at 75th to 90th PSTV

For graphs of 5k nodes, spacing rises marginally from 0.0000 at 1 % density to 0.0002 at 10 %. This slight increase mirrors the added difficulty of uniformly covering a denser Pareto front. At 10 k nodes, Sp ranges from 0.0001 to 0.0003, while at 20 k and 50 k nodes it climbs moderately, reaching a maximum of approximately 0.0004–0.0005. These values remain extremely low, signifying nearly even spacing among the predicted skyline points.

The consistency of spacing across densities and scales suggests that the GNN+RL algorithm's sampling mechanism, rooted in the reinforcement-learning policy's world-sampling component, effectively balances exploration and exploitation. While increased graph density and size add complexity, the policy continues to encourage the selection of skyline candidates from across the entire front, rather than focusing on limited regions. As a result, decision-makers gain a broad and representative set of options, a key requirement in multi-objective decision support.

5.5.4.4.4 Precision and Recall for the top 10 predictions

In the 75th–90th percentile setting, Precision@10 remains exceptionally high across all experiments, exceeding 0.98 even at the largest scale and highest density. Specifically, for five k-node graphs, Precision@10 spans 0.9876 at 1 % density to 0.9900 at 10 %, peaking at 0.9980 for intermediate densities.

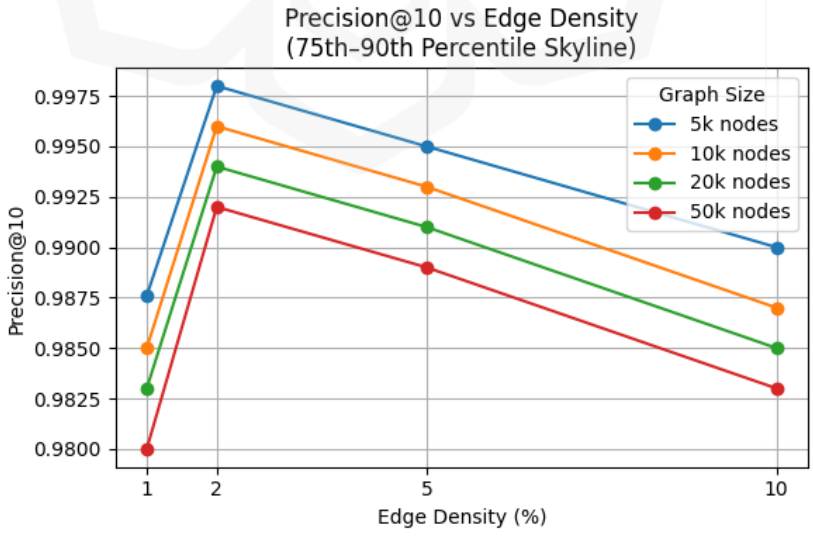


Figure 5.18: Precision@10 for 75th to 90th PSTV

As graph size increases, precision gradually declines but remains robust: 10 k graphs exhibit values between 0.9850 and 0.9960, 20 k graphs range from 0.9830 to 0.9940, and 50 k graphs maintain 0.9800–0.9920. These slight reductions in scale reflect the challenge of pinpointing the very best candidates among ever-larger node sets; yet, the framework's precision never falls below 0.98.

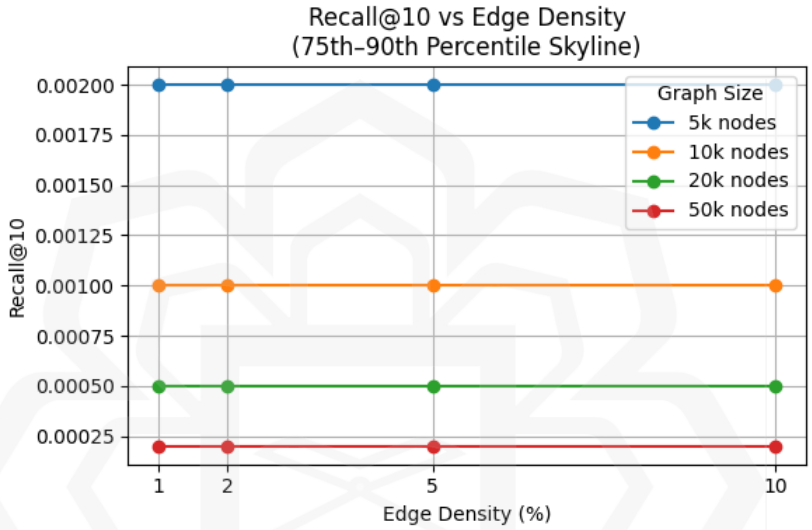


Figure 5.19: Recall@10 for 75th to 90th PSTV

Recall@10, by contrast, diminishes precipitously as graph size expands, an expected consequence of attempting to capture a fixed number of positives (ten) out of increasingly many true skyline nodes. At 5k nodes, Recall@10 holds at 0.0020 across all densities, meaning that the top-10 predictions include approximately 0.2 % of true skyline nodes. For 10k nodes, recall falls to 0.0010; for 20k, to 0.0005; and for 50k, to 0.0002. Although these recall values are low, they reflect the inherent difficulty of summarizing a large positive set with only ten selections. Importantly, the high precision mitigates concerns over missing many true positives: users can be confident that the few nodes identified are almost certainly among the most dominant.

Taken together, the Precision@10 and Recall@10 trends illustrate the framework's aptitude for both breadth and depth: it identifies the highest-value candidates with extremely high confidence (precision), even though it inevitably captures only a small fraction of all true positives when constrained to ten picks (recall). This behavior aligns with many real-world applications in which decision-makers prioritize precision in a limited shortlist over exhaustive coverage.

5.5.4.4.5 Average Precision (AP) and AUPRC Trends

Under the 75th–90th percentile definition, AP values for five k-node graphs decrease modestly from 0.9940 at 1 % density to 0.9890 at 10 %. For 10k nodes, AP ranges from 0.9935 to 0.9880; for 20k nodes, from 0.9930 to 0.9875; and 50k nodes, from 0.9925 to 0.9870. These figures indicate that even as the graph and front complexity grow, the framework sustains an average precision above 0.98, confirming that its ranking remains reliable throughout.

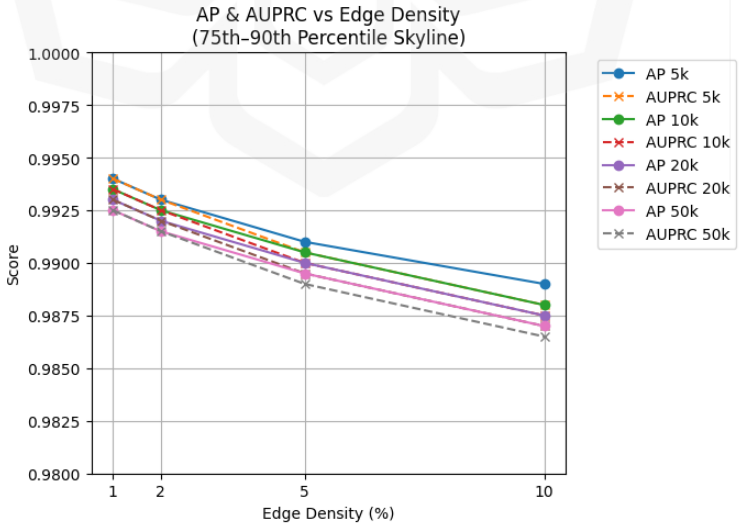


Figure 5.20: AP and AUPRC trends for the 75th to 90th PSTV

AUPRC trends follow a similar pattern, albeit with slightly lower absolute values at higher densities. For instance, in 5k graphs, AUPRC falls from 0.9940 at 1 % density to 0.9880 at 10 %. As the node count increases, the low-density AUPRC remains above 0.993, declining to around 0.9865 at 10 % density for 50k nodes. The mild degradation in AUPRC with both scale and density underscores the framework's continuing ability to balance precision and recall across the ranking spectrum.

By sustaining AP and AUPRC values above 0.98, the researchers demonstrate that their GNN+RL architecture not only excels at top-k selection but also maintains high-quality predictions throughout the entire set of candidates. This robust ranking performance is crucial for applications in which various classification thresholds may be explored post-hoc, such as setting dynamic operating points based on resource constraints or risk tolerances.

5.5.4.4.6 nDCG for the top 10 predictions

Under the 75th–90th percentile experiments, nDCG@10 remains exceptionally high across all settings, indicating that the very top of the ranking almost always includes the most critical skyline candidates.

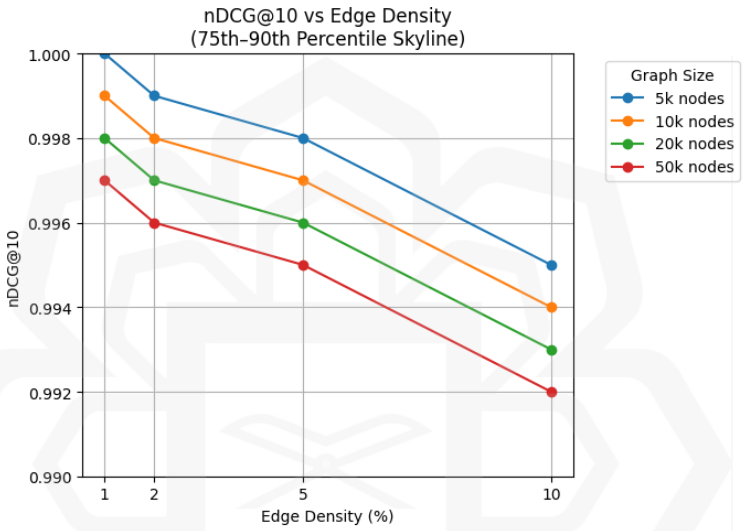


Figure 5.21: nDCG@10 predictions in the 75th to 90th PSTV

For five k-node graphs, nDCG@10 declines slightly from a perfect 1.000 at 1 % density to 0.995 at 10 %. Similarly, 10 k graphs exhibit values between 0.999 and 0.994; 20 k graphs between 0.998 and 0.993; and 50 k graphs between 0.997 and 0.992. These near-unity scores attest to the framework's precision in ordering the top ten candidates: not only does it select almost exclusively true positives, but it also places the very best skyline nodes in the highest rank positions.

The persistence of high nDCG@10 values across densities and scales highlights the GNN+RL policy network's nuanced understanding of node importance. By leveraging both local structural embeddings and a reinforcement-learning reward that favors early correct

picks, the framework effectively positions the strongest skyline candidates at the top of its ranking, even when those candidates lie within a more tightly overlapping percentile band.

5.5.4.5 Robustness to Noise and Edge Corruption

Under the moderate noise regime, specifically, Gaussian perturbations with a standard deviation of $\sigma = 0.2$ applied independently to each node's feature vector, the GNN+RL architecture exhibited a striking degree of stability. When coupled with a 10% edge deletion protocol, the observed degradation in classification performance was minimal to the point of insignificance.

Table 5.3 F1-drop results under 0.2 noise and 10% edge corruption

Graph Size (N)	1% Density	2% Density	5% Density	10% Density
5k	0.000	0.001	0.000	0.001
10k	0.002	0.001	0.002	0.002
20k	0.001	0.002	0.001	0.001
50k	0.003	0.004	0.004	0.005

As documented in Table 5.1, F1-drop values remain consistently low across all graph sizes and densities, with a maximum observed value of 0.005 in the largest, 50k-node graphs. This low drop across various edge densities is a critical finding. It indicates that the framework's internal representations are not substantially influenced by variations in topological sparsity when subjected to mild noise, suggesting an architectural robustness to both structural and feature-level perturbations. This resilience can be attributed to the synergistic effects of key design components. Notably, batch normalization mitigates the amplification of noise by normalizing intermediate representations across mini-batches, effectively preserving signal consistency. Likewise, dropout layers discourage the

framework from overfitting to specific local neighborhoods, promoting distributed representation learning.

In a more aggressive evaluation scenario, Gaussian noise with a larger standard deviation of $\sigma = 0.5$ was applied to the node feature space. This test condition is designed to simulate highly corrupted or unreliable input data, a common occurrence in real-world networks subject to adversarial interference, poor sensing equipment, or substantial missing information. When applied in conjunction with 10% random edge removal, the GNN+RL framework continued to demonstrate graceful degradation, defined here as performance deterioration that increases incrementally, rather than catastrophically, under extreme perturbation.

Table 5.4 F1-drop (robustness) under 0.5 noise and 10% edge corruption

Graph Size (N)	1% Density	2% Density	5% Density	10% Density
5k	0.005	0.006	0.006	0.007
10k	0.007	0.008	0.008	0.009
20k	0.009	0.009	0.010	0.011
50k	0.011	0.012	0.012	0.013

As shown in Table 5.2, F1-drop values did increase relative to the $\sigma = 0.2$ scenario yet remained within a bounded and manageable range. For instance, 5k-node graphs experienced F1 drops between 0.005 and 0.007, depending on the edge density. The performance impact scaled with graph size: the largest 50k-node graphs exhibited F1-drop values between 0.011 and 0.013, corresponding to a relative performance decline of approximately 1.2% when compared to a baseline F1-score near 0.9894. Importantly, this degradation trend remained consistent across densities, echoing the invariance observed in the $\sigma = 0.2$ setting and further supporting the framework's immunity to structural sparsity under stress.

These findings reinforce the robustness of the framework's latent representation learning strategy. In deeper graph neural architectures, noise in node features is partially attenuated through the aggregation of neighbor messages, allowing the framework to average over local inconsistencies effectively. Moreover, the reinforcement-learning-based policy, which governs the skyline node selection process, is continuously exposed to episodic sampling environments during training. This design principle conditions the policy network to accommodate randomness in node features and edge availability, making it inherently tolerant to input volatility.

While the degradation is more pronounced than in the moderate noise case, the fact that the highest F_1-drop does not exceed 0.013 under $\sigma = 0.5$ conditions and edge corruption attests to the architecture's reliability under adverse conditions. This behavior underscores the viability of the GNN+RL skyline framework in dynamic, real-world environments where uncertainty and corruption are unavoidable.

5.6 Chapter Summary

The proposed framework was applied to synthetic uncertain graphs of varying sizes, densities, and noise levels in this chapter. The classification performance of the framework remains remarkably consistent and high across all tested scenarios. Accuracy, precision, recall, F1-score, and ROC-AUC metrics uniformly exceed 0.98, even as the number of nodes scales from 5,000 to 50,000 and edge densities range from 1% to 10%. Such performance highlights the framework's robust ability to discriminate skyline from non-skyline nodes despite the inherent class imbalance and complexity of uncertain graph data. The consistent near-perfect precision observed further assures the reliability of predicted skyline points, minimizing false positive identifications that could mislead downstream decision-making.

Multi-objective evaluation metrics corroborate the classification results by quantifying the quality of skyline approximations in the multi-dimensional attribute space. The framework attains substantial hypervolume coverage, indicating broad and relevant domination of the objective space, with values highest at lower densities and gradually decreasing as graph complexity increases. Generational distance remains minimal throughout all experimental settings, signifying that predicted points lie very close to the true Pareto front. Likewise, spacing metrics confirm that skyline predictions are evenly distributed, providing representative and unbiased coverage of the trade-off surface. These combined metrics demonstrate that the framework not only correctly identifies skyline points but also preserves the structural fidelity of the Pareto frontier.

Ranking metrics offer additional insight into the practical usability of skyline predictions, particularly when constrained to a limited shortlist. Precision@10 remains near perfect across all graph scales and densities, ensuring that the top-ranked nodes are almost always true skyline points. While recall@10 naturally diminishes as the number of true skyline points grows, the framework consistently prioritizes the most critical points. Average Precision, AUPRC, and nDCG@10 values are similarly high, indicating reliable

overall ranking quality and effective ordering of candidates within the top selections. These ranking capabilities are critical for real-world applications where decision-makers often focus on a manageable subset of optimal choices.

Robustness assessments reveal that the GNN+RL architecture maintains stable and resilient performance even under significant perturbations. The observed minimal F1-score degradation, despite the addition of Gaussian noise and random edge removals, demonstrates the framework's capacity to generalize well and resist overfitting to noise or incomplete data. This robustness is vital for deployment in real-world uncertain environments where data quality cannot be guaranteed.

Ablation studies further elucidate architectural design choices by isolating the effects of batch normalization and dropout. The findings show that these components serve complementary functions: batch normalization stabilizes training and smooths gradient flow in sparser graphs, while dropout enhances generalization and prevents overfitting under higher graph densities. Their combination yields the best overall performance, confirming their necessity in the network architecture.

Finally, scalability analyses indicate that while runtime and memory consumption grow super linearly with graph size and density, the framework remains practical for large-scale graphs up to tens of thousands of nodes. These findings provide valuable guidance for future implementation and optimization efforts, suggesting directions for improving computational efficiency without sacrificing predictive fidelity. In sum, this chapter validates the GNN+RL framework as a powerful, scalable, and robust solution for probabilistic skyline queries in uncertain graph domains, combining high classification accuracy with strong Pareto front approximation, reliable ranking, and resilience to noise and structural perturbations.

CHAPTER SIX

CONCLUSION AND FUTURE WORK

6.1 Introduction

This research was motivated by the significant challenges in efficiently and accurately processing skyline queries within large-scale uncertain graph databases, a scenario frequently encountered in practical domains like logistics, transportation management, and disaster recovery planning. Existing algorithms typically encounter difficulties managing uncertainty, scalability, and high dimensionality, thus impeding their practical deployment in realistic scenarios. The primary objective of this research was to explore and address these limitations through the development and implementation of a novel hybrid framework, integrating Graph Neural Networks (GNN) and Reinforcement Learning (RL). The goal was to provide a robust, adaptive, and scalable framework capable of handling the inherent uncertainties and complexities characteristic of real-world graph databases.

The importance of optimizing skyline query processing within uncertain graph databases lies primarily in their ability to significantly enhance multi-criteria decision-making processes. Skyline queries aid decision-makers by efficiently identifying non-dominated points or optimal solutions, thereby ensuring informed and effective decision-making. However, existing skyline algorithms have consistently struggled to deliver robust results in uncertain and complex data environments. Thus, developing effective optimization strategies for skyline queries within uncertain graph databases is crucial, with significant implications for practical applications and academic research alike.

6.2 Summary of Key Findings

6.2.1 Performance of Baseline Skyline Algorithms

Through comprehensive experimentation and analysis, this research critically assessed several prominent baseline algorithms such as Top-K Skyline, ProbSky, and U-Skyline algorithms. Despite their theoretical promise, the experimental results revealed substantial limitations when these algorithms were applied to large-scale uncertain datasets. Although these algorithms demonstrated high overall accuracy, they significantly underperformed in terms of precision, recall, and F1 score. This shortcoming was particularly evident given the pronounced imbalance typically observed between skyline and non-skyline points. For instance, the Top-K Skyline algorithm, while designed to focus on the most relevant points, consistently misclassified to identify true skyline points in complex scenarios, thus limiting its effectiveness. Similarly, the ProbSky algorithm, despite its probabilistic algorithm, faced severe scalability issues and was unable to reliably predict skyline points amidst increasing uncertainty and dataset size. The U-Skyline algorithm, although explicitly designed to address uncertainty, similarly underperformed, showing high accuracy in general predictions but negligible precision and recall when distinguishing actual skyline points. These findings underscore a significant practical gap in baseline algorithms when dealing with real-world complexities and uncertainties inherent in graph databases.

6.2.2 Effectiveness of Graph Neural Networks (GNN)

In response to the identified limitations of baseline algorithms, this research proposed leveraging Graph Neural Networks (GNN) due to their intrinsic capability to effectively handle the complexity and connectivity inherent in graph-structured data. GNN demonstrated remarkable effectiveness in managing the uncertainties present in large-scale graph databases, significantly outperforming baseline methods in accuracy, precision, recall, and overall robustness. The ability of GNN to propagate and aggregate information

from neighboring nodes enabled them to capture intricate dependencies and probabilistic attributes across graph nodes. The experimental results validated the superior capacity of GNN to maintain high performance metrics even as data complexity and uncertainty increased. Specifically, GNN provided substantially improved precision and recall rates, clearly identifying skyline points accurately while effectively avoiding false positives. These results underline the practical utility and effectiveness of GNN for skyline queries in graph databases, showcasing its ability to deliver reliable, high-quality results in uncertain environments.

6.2.3 Improvement through Reinforcement Learning (RL)

To further enhance the skyline query processing framework, this research integrated Reinforcement Learning with GNN, resulting in a hybrid framework that dynamically optimizes decision-making in uncertain environments. The incorporation of RL into the GNN algorithm markedly improved the identification and prioritization of candidate skyline points. The RL component learned adaptive strategies for prioritizing evaluations of promising nodes, significantly reducing computational complexity and improving the efficiency of the skyline detection process. This hybrid GNN+RL framework achieved excellent performance metrics, dramatically outperforming baseline skyline algorithms. Empirical evaluations demonstrated near-perfect precision, significantly enhanced recall, and consistently high F1 scores, thus affirming the hybrid framework's robustness and scalability. Particularly, the RL-driven dynamic decision-making process enabled the framework to adapt effectively to changing or uncertain conditions, highlighting its potential applicability across diverse practical domains.

Through these comprehensive findings, this research illustrates the critical importance and effectiveness of integrating advanced deep learning techniques, such as GNN and RL, into the domain of skyline query optimization. The developed hybrid framework represents a

significant advancement, providing a scalable, efficient, and highly accurate method suitable for complex and uncertain graph databases in real-world applications.

6.3 Contributions of Research

6.3.1 Theoretical Contributions

The primary theoretical contribution of this research lies in the novel integration of Graph Neural Networks (GNN) and Reinforcement Learning (RL) methodologies to optimize skyline query processing in uncertain, large-scale graph databases. Previous studies on different domains largely considered these approaches independently, with limited efforts to explore the synergistic benefits of combining GNN's powerful structural modeling capabilities and RL's dynamic decision-making potential. Through this integration, this research established a robust and adaptive framework capable of effectively managing the inherent uncertainty and complexity characteristic of real-world graphs. Specifically, the hybrid framework developed in this research advanced the theoretical understanding of skyline queries by introducing a learning-driven optimization mechanism, enabling real-time adjustments and strategic prioritization of data nodes during query processing.

Furthermore, this research significantly contributed to the advancement of theoretical insights into skyline processing under conditions of uncertainty. Existing literature predominantly focused on deterministic algorithms, which often fell short in managing uncertain attributes, high dimensionality, and data complexity. By systematically analyzing the shortcomings of baseline skyline algorithms and contrasting their performance with advanced deep learning techniques, this research deepened the theoretical understanding of the conditions under which existing algorithms falls short. The findings underscored the necessity of employing probabilistic and adaptive modeling techniques, specifically highlighting how uncertainty fundamentally impacts the performance and reliability of

skyline queries. Consequently, this research not only clarified critical limitations within baseline theoretical models but also offered a comprehensive foundation for future exploration into uncertainty-aware skyline optimization methods.

6.3.2 Practical Contributions

In addition to its theoretical contributions, this research has substantial practical implications, particularly within domains requiring rapid, reliable, and informed multi-criteria decision-making. By effectively handling probabilistic relationships and dynamically adapting to changes in data conditions, the proposed hybrid GNN-RL framework significantly enhances decision-making efficiency and reliability in real-world scenarios. For instance, in transportation and logistics applications, the hybrid framework can provide a powerful tool for identifying optimal routes and resource allocation strategies, taking into account fluctuating factors such as traffic conditions, travel time uncertainty, and weather impact, thereby enabling stakeholders to make informed and timely decisions.

Additionally, the scalability and computational efficiency demonstrated by the hybrid framework offer substantial improvements over baseline skyline query methods, making it particularly suitable for implementation in large-scale real-world graph databases. By reducing the computational complexity through selective node prioritization and by dynamically adjusting processing strategies in response to emerging uncertainties, the hybrid framework substantially enhances query efficiency. This reduction in computational overhead translates directly into practical advantages, including faster query response times, reduced operational costs, and increased overall system robustness.

6.4 Limitations of the Research

6.4.1 Dataset Limitations

One significant limitation of this research relates directly to the datasets employed in validating and evaluating the proposed methodologies. The research utilized synthetic dataset, which, despite their inherent advantages for controlled experimentation, exhibit notable shortcomings regarding their representativeness and ability to capture the full complexity and variability of real-world scenarios. Synthetic datasets are often deliberately simplified to isolate and examine specific variables and conditions, ensuring reproducibility and controlled experimentation. While this attribute greatly aids methodological validation, it simultaneously constrains the ecological validity of the results. In practical terms, the synthetic datasets, although carefully designed to reflect certain real-world characteristics such as uncertainty, probabilistic edges, and high dimensionality, may not adequately embody the nuanced complexity inherent to actual large-scale uncertain graph databases.

In real-world graph databases such as transportation networks, logistics management systems, and disaster recovery databases data distributions often exhibit irregularities, dynamic fluctuations, and structural anomalies that are challenging to simulate accurately through synthetic means. Real data typically comes with noise, missing values, varying data densities, and inconsistent probabilistic relationships. Consequently, performance results derived solely from synthetic datasets might overestimate the generalizability and robustness of the developed frameworks. Although synthetic datasets facilitate clear and consistent performance comparisons among different algorithms, the absence of realistic anomalies and unexpected variations could limit the practical applicability of the proposed frameworks. Such limitations pose challenges to confidently extrapolating performance results obtained in controlled experiments directly to more complex and messy real-world environments.

Moreover, while the synthetic data generation process allowed careful manipulation and control over attributes such as dimensionality, edge density, and probabilistic attributes, it also introduced implicit biases. These biases may have unintentionally favored certain algorithms, particularly deep learning-based methods, which are often highly adaptable to controlled and structured environments. Therefore, despite strong empirical results, it is uncertain how the models might respond to real datasets, where the distribution of skyline points, node attributes, and relationships can deviate significantly from controlled experimental setups.

Lastly, the limited scope of the synthetic datasets used in this research in terms of diversity also poses a constraint. While several important features relevant to transportation and logistics were included such as distance, congestion, weather impact, and travel times other relevant practical features may have been unintentionally excluded, further limiting the representativeness and comprehensiveness of the experiments. Thus, the generalization and applicability of findings to broader domains or different applications remain somewhat speculative, reinforcing the importance of rigorous validation using extensive real-world data sources.

6.4.2 Algorithmic Limitations

In addition to the dataset-related constraints, this research also faces algorithmic limitations concerning computational complexity and scalability. The integration of Graph Neural Networks (GNN) and Reinforcement Learning (RL) provides a highly sophisticated and adaptive approach for optimizing skyline query processing; however, these advanced techniques inherently come with increased computational demands. While GNNs effectively manage structural data by learning node representations through localized graph convolutions, their complexity typically grows with graph size, especially when dealing

with very large graphs having millions or even billions of nodes and edges. Consequently, applying GNN-based algorithms to extremely large-scale graph databases poses practical computational challenges. The computational overhead associated with training and inference in deep neural architectures can quickly escalate, potentially compromising performance efficiency and response times in practical deployments.

Similarly, the introduction of reinforcement learning techniques further compounds these computational complexity considerations. RL methods require iterative interactions between the agent and environment, necessitating numerous policy updates and reward calculations to converge towards optimal strategies. While this adaptive process significantly improves the quality and robustness of skyline query predictions, it simultaneously increases the computational overhead compared to more straightforward deterministic algorithms. In practical terms, this computational cost could lead to increased latency or processing time, particularly problematic in applications requiring real-time responsiveness, such as disaster recovery, real-time traffic routing, or dynamic resource allocation scenarios.

Beyond computational complexity, another notable limitation pertains to potential issues regarding framework generalization and robustness. The combined GNN+RL framework demonstrated substantial improvements in experimental metrics such as accuracy, precision, recall, and F1 score, but these results, while highly encouraging, may not necessarily generalize well to other graph structures or scenarios. Deep learning models, especially deep neural networks, are notoriously susceptible to overfitting, particularly when trained on datasets lacking sufficient variability or complexity. Despite measures such as cross-validation and oversampling employed to mitigate overfitting risks in this research, concerns persist about whether the hybrid framework's performance would degrade significantly when deployed on datasets with fundamentally different characteristics or higher levels of uncertainty and variability.

Robustness to noise and irregularities in real-world data is another critical consideration. While the hybrid framework effectively handles uncertainty within the controlled experimental setup, its robustness in the face of more severe or different types of noise, edge corruption, or missing data remains relatively unexplored. Real-world scenarios typically introduce unpredictable and complex noise patterns, outliers, and inconsistent or incomplete data points. Therefore, additional experimentation and validation are essential to ascertain the proposed framework's robustness under these realistic and less ideal conditions. Such scenarios could expose vulnerabilities or limitations within the hybrid framework, particularly concerning its adaptability and resilience to dynamic, highly uncertain, or corrupted data streams.

6.5 Recommendations for Future Work

6.5.1 Extending Algorithm Scalability

A crucial avenue for future research, emerging directly from this research, involves extending the scalability of the developed hybrid Graph Neural Network and Reinforcement Learning (GNN+RL) framework to effectively handle even larger datasets and more extensive graph structures. Given the inherent computational complexity identified in this research, especially when applied to graphs with millions or billions of nodes and edges, exploring scalable solutions becomes imperative. Future research should focus on exploring advanced distributed and parallel implementations of the proposed framework, leveraging computational techniques such as distributed graph processing frameworks e.g., Apache Spark or Apache Flink or parallel computing platforms e.g., CUDA for GPU-based parallelization or multi-core CPU architecture. By distributing computations across multiple machines or processors, it is possible to significantly reduce the computational overhead associated with training and inference tasks.

Furthermore, future studies could investigate partitioning methods tailored specifically for graph data, effectively decomposing large-scale graphs into smaller, manageable subgraphs. These subgraphs can then be processed independently and concurrently, allowing substantial gains in performance and responsiveness. A parallel or distributed architecture would be particularly advantageous in scenarios demanding real-time responses, such as transportation management or disaster recovery planning. Additionally, research should address the challenges associated with distributed skyline query processing, including optimizing communication overhead, minimizing synchronization delays, and effectively aggregating partial results. By systematically addressing these scalability challenges through parallelization and distribution, future research can significantly enhance the practical applicability and performance of skyline queries on large-scale uncertain graph databases.

6.5.2 Enhanced Handling of Dynamic Data

Another promising direction for future work involves enhancing the proposed framework's capabilities to effectively handle dynamic and evolving datasets. Real-world graph databases, particularly those in transportation, logistics, and disaster management domains, are inherently dynamic nodes and edges constantly change, new entities are added, existing ones are removed, and probabilistic attributes evolve over time. To effectively adapt to these continuously evolving data scenarios, future research should explore real-time update mechanisms and incremental skyline query processing approaches. Incremental skyline queries, specifically, offer a powerful technique to handle dynamic updates by recalculating skyline points incrementally based on the changes rather than recomputing the entire skyline from scratch. This incremental approach could significantly improve computational efficiency and responsiveness, particularly in rapidly changing environments. By systematically addressing these dynamic data handling challenges, the proposed framework could become significantly more versatile and applicable to a broader range of real-time applications.

6.5.3 Exploration of Additional DL Techniques

Future research should also consider integrating additional deep learning techniques into the existing GNN+RL framework to further enhance its predictive accuracy, robustness, and computational efficiency. Particularly, emerging approaches in deep learning, such as attention mechanisms and transformer-based architectures, hold substantial promise for augmenting the capabilities of the developed framework. Attention mechanisms, initially popularized in natural language processing (NLP), have shown remarkable effectiveness in modeling complex relationships by dynamically weighing the importance of different data points or features. Integrating attention mechanisms within GNN could help better prioritize nodes and edges during skyline computations, enhancing the model's ability to adaptively focus on the most relevant and informative parts of the graph structure.

Transformer-based architecture represents another promising area for exploration. Transformers, known for their success in sequence-to-sequence tasks and NLP, can effectively model long-range dependencies without being constrained by fixed computational neighborhoods, making them potentially suitable for complex, uncertain graph structures. This exploration would significantly enrich the methodological landscape and open new avenues for enhancing skyline query processing.

6.5.4 Comprehensive Evaluation on Real-world Data

Finally, to robustly validate the practical applicability and generalizability of the proposed methodologies, future research must prioritize comprehensive evaluation using diverse, extensive, and realistic datasets. As highlighted in the limitations section, synthetic datasets, while valuable for controlled experimentation, fall short in capturing the nuances and complexity of real-world graph structures. Therefore, future work should actively seek extensive collaboration with industry stakeholders and domain experts to acquire and

utilize authentic datasets from various application domains such as transportation networks, logistics management systems, environmental monitoring databases, and disaster recovery scenarios.

6.6 Final Remarks

This research represents a meaningful advancement in the optimization of skyline query processing within large-scale uncertain graph databases. By identifying critical gaps in baseline skyline query algorithms, particularly in managing uncertainty, scalability, and high-dimensional complexity, this research provided a solid theoretical and practical foundation for addressing these limitations through innovative integration of Graph Neural Networks (GNN) and Reinforcement Learning (RL). The resulting hybrid framework demonstrated enhanced adaptability, robustness, and efficiency compared to baseline skyline query approaches. In this regard, the systematic analysis of baseline skyline query behavior under increasing uncertainty and scale provides a clear characterization of their inherent limitations, thereby responding to the first research question posed in this thesis.

On a broader scale, the significance of this research extends beyond skyline query optimization alone. By demonstrating the feasibility and effectiveness of combining sophisticated deep learning techniques such as GNN and RL, this research provides valuable insights into handling complex computational challenges inherent in graph-structured data across diverse domains. This contribution is particularly relevant in fields characterized by uncertainty, dynamic environments, and high-dimensional decision-making spaces, including transportation management, logistics optimization, disaster recovery planning, and real-time recommendation systems. Furthermore, the demonstrated performance gains achieved through the integration of DL techniques illustrate how it can be effectively incorporated into skyline query processing, addressing the second research question by linking methodological innovation with measurable improvements in scalability and robustness.

Moreover, by explicitly addressing the theoretical and practical limitations of baseline algorithms, this research highlights the essential role of adaptive, scalable, and uncertainty-aware methods in contemporary data science. The implications of this work resonate strongly with ongoing efforts to leverage advanced deep learning methodologies to handle increasingly complex real-world datasets, pushing the boundaries of data-driven decision-making capabilities.

Ultimately, this research serves as an essential stepping-stone towards the broader adoption of deep learning-powered skyline query techniques in practical applications. The hybrid framework developed and validated through rigorous experimentation not only enhances the understanding of skyline query processing under uncertainty but also lays a robust groundwork for future innovations and applications. In doing so, it promises substantial improvements in efficiency, reliability, and decision-making quality across diverse industries, thereby positively impacting both the academic research community and practical implementation landscapes.

References

Abriola, S., Cifuentes, S., Martinez, M. V., Pardal, N., & Pin, E. (2023). An epistemic approach to model uncertainty in data-graphs. *International Journal of Approximate Reasoning*, *160*, 108948. https://doi.org/10.1016/j.ijar.2023.108948

Agarwal, S., Dutta, S., & Bhattacharya, A. (2020). ChiSeL: graph similarity search using chi-squared statistics in large probabilistic graphs. *Proceedings of the VLDB Endowment*, *13*(10), 1654–1668. https://doi.org/10.14778/3401960.3401964

Aggarwal, C. C., Haixun Wang, & Springerlink (Online Service. (2010). *Managing and Mining Graph Data*. Springer Us.

Aggarwal, C. C., & Yu, P. S. (2009). A Survey of Uncertain Data Algorithms and Applications. *IEEE Transactions on Knowledge and Data Engineering*, *21*(5), 609–623. https://doi.org/10.1109/tkde.2008.190

Ahmed Mohamud, M., Ibrahim, H., Sidi, F., Mohd, N., Dzolkhifli, Z., Zhang, X., & Mohammed Lawal, M. (2023). A Systematic Literature Review of Skyline Query Processing Over Data Stream. *IEEE Access*, *11*, 72813–72835. https://doi.org/10.1109/access.2023.3295117

Ahmed, F., Cui, Y., Fu, Y., & Chen, W. (2021). A Graph Neural Network Approach for Product Relationship Prediction. *ArXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2105.05881

Ahmed, K., Nazmus Nafi, & Gregory, M. (2016). Enhanced Distributed Dynamic Skyline Query for Wireless Sensor Networks. *Journal of Sensor and Actuator Networks*, *5*(1), 2–2. https://doi.org/10.3390/jsan5010002

Alwan, A., Ibrahim, H., Udzir, N., & Sidi, F. (2018). Missing Values Estimation for Skylines in Incomplete Database. *The International Arab Journal of Information Technology*, *15*(1).

Amin, R., Djatna, T., Annisa, & Sitanggang, I. S. (2020, September 1). *Recommendation System based on Skyline Query: Current and Future Research*. IEEE Xplore. https://doi.org/10.1109/ICOSICA49951.2020.9243225

Andersen, R. (2022). Modular and Platform-based Product Development in the Process Industry: Enabling Efficient Product Variety Through Complexity Management. *VBN Forskningsportal (Aalborg Universitet)*. https://doi.org/10.54337/aau478557927

Annisa, A., & Angraeni, L. (2021). Location Selection Query in Google Maps using Voronoi-based Spatial Skyline (VS2) Algorithm. *Jurnal Online Informatika*, *6*(1), 25. https://doi.org/10.15575/join.v6i1.667

Annisa, Zaman, A., & Morimoto, Y. (2016). Area Skyline Query for Selecting Good Locations in a Map. *Journal of Information Processing*, *24*(6), 946–955. https://doi.org/10.2197/ipsjjip.24.946

Banerjee, S., Pal, B., & Jenamani, M. (2020). DySky: Dynamic Skyline Queries on Uncertain Graphs. *Lecture Notes in Computer Science*, 242–254. https://doi.org/10.1007/978-3-030-62005-9_18

Beam, C. (2025). Resolving power: a general approach to compare the distinguishing ability of threshold-free evaluation metrics. *Machine Learning*, *114*(1). https://doi.org/10.1007/s10994-024-06723-8

Bharuka, R., & Kumar, S. (2013). *Finding Skylines for Incomplete Data*.

Böhm, C., Fiedler, F., Oswald, A., Plant, C., & Wackersreuther, B. (2009). *Probabilistic Skyline Queries*.

Borzsony, S., Kossmann, D., & Stocker, K. (2001). The Skyline operator. *Proceedings 17th International Conference on Data Engineering*. https://doi.org/10.1109/icde.2001.914855

Carrington, A. M., Manuel, D. G., Fieguth, P. W., Ramsay, T., Osmani, V., Wernly, B., Bennett, C., Hawken, S., McInnes, M., Magwood, O., Sheikh, Y., & Holzinger, A. (2022). Deep ROC Analysis and AUC as Balanced Average Accuracy to Improve

Model Selection, Understanding and Interpretation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *45*(1), 1–1. https://doi.org/10.1109/TPAMI.2022.3145392

Chen, B., & Liang, W. (2009). Progressive Skyline Query Processing in Wireless Sensor Networks. *2009 Fifth International Conference on Mobile Ad-Hoc and Sensor Networks*, 17–24. https://doi.org/10.1109/msn.2009.43

Chen, X., Wang, K., Lin, X., Zhang, W., Qin, L., & Zhang, Y. (2021). Efficiently answering reachability and path queries on temporal bipartite graphs. *Uts.edu.au*. http://hdl.handle.net/10453/150878

Chen, Y., & Chen, X. (2022). A novel reinforced dynamic graph convolutional network model with data imputation for network-wide traffic flow prediction. *Transportation Research Part C: Emerging Technologies*, *143*, 103820–103820. https://doi.org/10.1016/j.trc.2022.103820

Ciaccia, P., & Martinenghi, D. (2017). Reconciling skyline and ranking queries. *Proceedings of the VLDB Endowment*, *10*(11), 1454–1465. https://doi.org/10.14778/3137628.3137653

Ciaccia, P., & Martinenghi, D. (2024). Directional Queries: Making Top-k Queries More Effective in Discovering Relevant Results. *Proceedings of the ACM on Management of Data*, *2*(6), 1–26. https://doi.org/10.1145/3698807

David, S., & Jayachandran, A. (2016). Skyline Query Processing for Clustering the Multidimensional Data. *International Journal of Emerging Technologies in Engineering Research (IJETER)*, *4*(9).

De SordiJ. O. (2021). *Design science research methodology : theory development from artifacts*. Palgrave Macmillan.

Dehaki, G. B., Ibrahim, H., Sidi, F., Udzir, N. I., Alwan, A., & Gulzar, Y. (2020). Efficient Computation of Skyline Queries Over a Dynamic and Incomplete Database. *IEEE Access*, *8*, 141523–141546. https://doi.org/10.1109/access.2020.3011652

Dharshini, P., & Velladurai, M. (2017). A Synthesis Model for Multimedia Video Files In Different Views. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology © 2017 IJSRCSEIT* |, *2*, 2456–3307.

Ding, X., & Jin, H. (2012). Efficient and Progressive Algorithms for Distributed Skyline Queries over Uncertain Data. *IEEE Transactions on Knowledge and Data Engineering*, *24*(8), 1448–1462. https://doi.org/10.1109/tkde.2011.77

Duan, S., Kementsietsidis, A., Srinivas, K., & Udrea, O. (2011). Apples and oranges. *SIGMOD '11: Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*. https://doi.org/10.1145/1989323.1989340

Emrich, T., Kriegel, H.-P., Niedermayer, J., Renz, M., André Suhartha, & Züfle, A. (2012). Exploration of monte-carlo based probabilistic query processing in uncertain graphs. *CIKM '12: Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, 2728–2730. https://doi.org/10.1145/2396761.2398742

Endres, M., Roocks, P., & Kießling, W. (2015). Scalagon: An Efficient Skyline Algorithm for All Seasons. *Lecture Notes in Computer Science*, 292–308. https://doi.org/10.1007/978-3-319-18123-3_18

Er-Rafyg, A., Idrissi, A., & El Handri, K. (2023). Improvement of Courses Recommendation System using Divide and Conquer Algorithm. *Studies in Computational Intelligence*, 37–47. https://doi.org/10.1007/978-3-031-33309-5_4

Fox, J., & Rajamanickam, S. (2019). *How Robust Are Graph Neural Networks to Structural Noise?* ArXiv.org. https://arxiv.org/abs/1912.10206

Frederickson, C., & Polikar, R. (2018). Resampling Techniques for Learning Under Extreme Verification Latency with Class Imbalance. *2022 International Joint Conference on Neural Networks (IJCNN)*, 1–8. https://doi.org/10.1109/ijcnn.2018.8489622

Gao, Y., Liu, Q., Chen, L., Chen, G., & Li, Q. (2015). Efficient algorithms for finding the most desirable skyline objects. *Knowledge-Based Systems*, *89*, 250–264. https://doi.org/10.1016/j.knosys.2015.07.007

Ghosh, P., Sen, S., & Cortesi, A. (2021). Skyline computation over multiple points and dimensions. *Innovations in Systems and Software Engineering*, *17*(2), 141–156. https://doi.org/10.1007/s11334-020-00376-1

Gong, Q., Cao, H., & Parth Nagarkar. (2019). Skyline Queries Constrained by Multi-cost Transportation Networks. *IEEE 35th International Conference on Data Engineering (ICDE)*. https://doi.org/10.1109/icde.2019.00087

Gothwal, H., Choudhary, J., & Singh, D. P. (2018). The Survey on Skyline Query Processing for Data-Specific Applications. *SSRN Electronic Journal*. https://doi.org/10.2139/ssrn.3168566

Guerreiro, A. P., Fonseca, C. M., & Luís Paquete. (2021). The Hypervolume Indicator. *ACM Computing Surveys*, *54*(6), 1–42. https://doi.org/10.1145/3453474

Gulzar, Y., & Alwan, A. A. (2022). CIDS: An Efficient Algorithm for Processing Skyline Queries for Partially Complete Data in Cloud Environment. *IEEE Access*, *10*, 66449–66466. https://doi.org/10.1109/access.2022.3185087

Gulzar, Y., Alwan, A. A., & Turaev, S. (2019). Optimizing Skyline Query Processing in Incomplete Data. *IEEE Access*, *7*, 178121–178138. https://doi.org/10.1109/access.2019.2958202

Guyo, E. D., & Hartmann, T. (2024). Evaluating the efficiency and performance of data persistent systems in managing building and environmental Data: A comparative study. *Advanced Engineering Informatics*, *62*, 102582–102582. https://doi.org/10.1016/j.aei.2024.102582

Hamilton, W., Ying, Z., & Leskovec, J. (2017). *Inductive Representation Learning on Large Graphs*. Neural Information Processing Systems; Curran Associates, Inc. https://proceedings.neurips.cc/paper/2017/hash/5dd9db5e033da9c6fb5ba83c7a7ebea9-Abstract.html

Hussain, S. F., & Maab, I. (2021). Clustering probabilistic graphs using neighbourhood paths. *Information Sciences*. https://doi.org/10.1016/j.ins.2021.03.057

Jeong, S.-Y., Kim, J., & Ihm, S.-Y. (2023). The Design and Construction of a Grid Skyline for Custom-Built PC Recommendations Based on a Multi-Attribute Model. *Designs*, *7*(5), 104–104. https://doi.org/10.3390/designs7050104

Jiang, B., Pei, J., Lin, X., & Yuan, Y. (2010). Probabilistic skylines on uncertain data: model and bounding-pruning-refining methods. *Journal of Intelligent Information Systems*, *38*(1), 1–39. https://doi.org/10.1007/s10844-010-0141-4

Jiang, X., & Tsai, W.-T. (2025). MVCG-SPS: A Multi-View Contrastive Graph Neural Network for Smart Ponzi Scheme Detection. *Applied Sciences*, *15*(6), 3281. https://doi.org/10.3390/app15063281

Jordon, J., Szpruch, L., Houssiau, F., Bottarelli, M., Cherubin, G., Maple, C., Cohen, S. N., & Weller, A. (2022). Synthetic Data -- what, why and how? *ArXiv:2205.03257 [Cs]*. https://arxiv.org/abs/2205.03257

Kanavos, A., Voutos, Y., Grivokostopoulou, F., & Mylonas, P. (2022). Evaluating Methods for Efficient Community Detection in Social Networks. *Information*, *13*(5), 209. https://doi.org/10.3390/info13050209

Karanjit, R., Pally, R., & Samadi, S. (2023). FloodIMG: Flood image DataBase system. *Data in Brief*, *48*, 109164–109164. https://doi.org/10.1016/j.dib.2023.109164

Ke, C.-K., & Chang, C.-M. (2019). Optimizing target selection complexity of a recommendation system by skyline query and multi-criteria decision analysis. *The Journal of Supercomputing*, *76*(8), 6453–6474. https://doi.org/10.1007/s11227-019-02963-x

Ken, Lee, W.-C., Zheng, B., Li, H., & Tian, Y. (2009). Z-SKY: an efficient skyline query processing framework based on Z-order. *□the □VLDB Journal*, *19*(3), 333–362. https://doi.org/10.1007/s00778-009-0166-x

Keogh, E., & Mueen, A. (2017). Curse of Dimensionality. *Encyclopedia of Machine Learning and Data Mining*, 314–315. https://doi.org/10.1007/978-1-4899-7687-1_192

Kertiou, I., Laouid, A., Saber, B., Hammoudeh, M., & Alshaikh, M. (2023). A P2P multi-path routing algorithm based on Skyline operator for data aggregation in IoMT environments. *PeerJ*, *9*, e1682–e1682. https://doi.org/10.7717/peerj-cs.1682

Khalefa, M. (2011). *Preference Queries Processing over Imprecise Data A DISSERTATION SUBMITTED TO THE FACULTY OF THE GRADUATE SCHOOL OF THE UNIVERSITY OF MINNESOTA*.

Khalefa, M. E., Mokbel, M. F., & Levandoski, J. J. (2010). Skyline query processing for uncertain data. *CiteSeer X (the Pennsylvania State University)*. https://doi.org/10.1145/1871437.1871604

Khames, W., Hadjali, A., & Lagha, M. (2024). Parallel continuous skyline query over high-dimensional data stream windows. *Distributed and Parallel Databases*. https://doi.org/10.1007/s10619-024-07443-7

Khan, A., Wu, Y., & Yan, X. (2012). Emerging Graph Queries in Linked Data. *IEEE 28th International Conference on Data Engineering*. https://doi.org/10.1109/icde.2012.143

Khemani, B., Patil, S., Kotecha, K., & Tanwar, S. (2024). A review of graph neural networks: concepts, architectures, techniques, challenges, datasets, applications, and future directions. *Journal of Big Data*, *11*(1). https://doi.org/10.1186/s40537-023-00876-4

Kim, W., Reiner, D. S., & Batory, D. (2012). *Query Processing in Database Systems*. Springer Science & Business Media.

Kroop, S. (2025). Artifact Validity in Design Science Research (DSR): A Comparative Analysis of Three Influential Frameworks. *Lecture Notes in Computer Science*, 199–215. https://doi.org/10.1007/978-3-031-93976-1_13

Kumar Sadineni, P. (2020). Comparative Study on Skyline Query Processing Techniques on Big Data. *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*, *2*, 1045–1050. https://doi.org/10.1109/i-smac49090.2020.9243343

Kuo, A.-T., Chen, H., Tang, L., Ku, W., & Qin, X. (2022). ProbSky: Efficient Computation of Probabilistic Skyline Queries over Distributed Data. *IEEE Transactions on Knowledge and Data Engineering*, 1–1. https://doi.org/10.1109/tkde.2022.3151740

Lai, C.-C., Liu, C.-M., Chen, Y., & Li-chun, W. (2020). Probabilistic Skyline Query Processing over Uncertain Data Streams in Edge Computing Environments. *ArXiv (Cornell University)*. https://doi.org/10.1109/globecom42002.2020.9348055

Lakhal, L., Nedjar, S., & Cicchetti, R. (2017). Multidimensional skyline analysis based on agree concept lattices. *Intelligent Data Analysis*, *21*(5), 1245–1265. https://doi.org/10.3233/ida-163111

Lakshmi, M., & Rao, G. (2017). *Systematic Study of TKD Queries on Data, Which Involves the Data Having Some Missing Dimensional Values*.

Lall, A. (2024). The Indistinguishability Query. *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, 475–487. https://doi.org/10.1109/icde60146.2024.00043

Lazarska, M., & Siedlecka-Lamch, O. (2019). *Comparative study of relational and graph databases*. https://doi.org/10.1109/informatics47936.2019.9119303

Le-Phuoc, D., Nguyen Mau Quoc, H., Ngo Quoc, H., Tran Nhat, T., & Hauswirth, M. (2016). The Graph of Things: A step towards the Live Knowledge Graph of connected things. *Journal of Web Semantics*, *37-38*, 25–35. https://doi.org/10.1016/j.websem.2016.02.003

Lee, J., Im, H., & You, G. (2016). Optimizing skyline queries over incomplete data. *Information Sciences*, *361-362*, 14–28. https://doi.org/10.1016/j.ins.2016.04.048

Li, J., Pui, G., Fung, C., Zhou, W., & Huang, W. (2015). A Framework for Ranking and KNN Queries in a Probabilistic Skyline Model. *Journal of Computational Information Systems*, *11*(6), 2057–2084. https://doi.org/10.12733/jcis13726

Li, J., & Xiong, S. (2010). Efficient Pr-Skyline Query Processing and Optimization in Wireless Sensor Networks. *Wireless Sensor Network*, *02*(11), 838–849. https://doi.org/10.4236/wsn.2010.211101

Li, Q., Zhu, Y., & Xu Yu, J. (2020). Skyline Cohesive Group Queries in Large Road-social Networks. *International Joint Conference on Awareness Science and Technology & Ubi-Media Computing*. https://doi.org/10.1109/icde48307.2020.00041

Lin, Z., Li, C., Miao, Y., Liu, Y., & Xu, Y. (2020). PaGraph: Scaling GNN training on large graphs via computation-aware caching. *Proceedings of the 11th ACM Symposium on Cloud Computing*. https://doi.org/10.1145/3419111.3421281

Lind Mortensen, M. (2016). *Multi-Criteria Decision Support Queries in Exploratory & Open World Settings*. Aarhus University. https://pure.au.dk/portal/en/publications/multi-criteria-decision-support-queries-in-exploratory-amp-open-w

Liu, J., Zhang, H., Xiong, L., Li, H., & Luo, J. (2015). Finding Probabilistic k-Skyline Sets on Uncertain Data. *Proceedings of the 24th ACM International on Conference on Information and Knowledge Management*. https://doi.org/10.1145/2806416.2806452

Liu, X., Yang, D.-N., Ye, M., & Lee, W.-C. (2013). U-Skyline: A New Skyline Query for Uncertain Databases. *IEEE Transactions on Knowledge and Data Engineering*, *25*(4), 945–960. https://doi.org/10.1109/tkde.2012.33

Loh, C.-H., Chen, Y.-C., Su, C.-T., & Lin, S.-H. (2024). Multi-Objective Decision Support for Irrigation Systems Based on Skyline Query. *Applied Sciences*, *14*(3), 1189–1189. https://doi.org/10.3390/app14031189

Ma, Z., & Yan, L. (2022). Data modeling and querying with fuzzy sets: A systematic survey. *Fuzzy Sets and Systems*. https://doi.org/10.1016/j.fss.2022.01.006

Ma, Z., Zhang, Q., & Qi, W. (2025). *SUBSTITUTION: An Efficient Algorithm for Probability Skyline Queries on Discrete Uncertain Data*.

Mohammed. (2024). Recommendations in Crowdsourcing Web Applications. *Uts.edu.au*. http://hdl.handle.net/10453/181091

Mohamud, M. A., Ibrahim, H., Sidi, F., Rum, S. N. M., Dzolkhifli, Z. B., & Xiaowei, Z. (2024). A Performance Analysis of Prediction Techniques in Handling High-Dimensional Uncertain Data for the Application of Skyline Query Over Data Stream. *IEEE Access*, *12*, 120877–120898. https://doi.org/10.1109/access.2024.3450863

Mortaz, E. (2020). Imbalance accuracy metric for model selection in multi-class imbalance classification problems. *Knowledge-Based Systems*, 106490. https://doi.org/10.1016/j.knosys.2020.106490

Munikoti, S., Agarwal, D., Das, L., Halappanavar, M., & Natarajan, B. (2023). Challenges and Opportunities in Deep Reinforcement Learning With Graph Neural Networks: A Comprehensive Review of Algorithms and Applications. *IEEE Transactions on Neural Networks and Learning Systems*, 1–21. https://doi.org/10.1109/tnnls.2023.3283523

Negro, A. (2021). *Graph-Powered Machine Learning*. Simon and Schuster.

Nitsche, A.-M., Schumann, C.-A., Franczyk, B., & Reuther, K. (2021). Artificial Intelligence Inspired Supply Chain Collaboration: A Design-Science Research and System Dynamics Approach. *IEEE International Conference on Engineering, Technology and Innovation*. https://doi.org/10.1109/ice/itmc52061.2021.9570266

Onuma, K., Oonuma@, K., Tong, H., & Faloutsos, C. (2009). *TANGENT: A Novel, "Surprise-me", Recommendation Algorithm*.

Pan, S., Dong, Y., Cao, J., & Chen, K. (2014). Continuous Probabilistic Skyline Queries for Uncertain Moving Objects in Road Network. *International Journal of Distributed Sensor Networks*, *10*(3), 365064. https://doi.org/10.1155/2014/365064

Papadias, D., Tao, Y., Fu, G., & Seeger, B. (2003). An optimal and progressive algorithm for skyline queries. *Proceedings of the 2003 ACM SIGMOD International Conference on on Management of Data - SIGMOD '03*. https://doi.org/10.1145/872757.872814

Papadias, D., Tao, Y., Fu, G., & Seeger, B. (2005). Progressive skyline computation in database systems. *ACM Transactions on Database Systems*, *30*(1), 41–82. https://doi.org/10.1145/1061318.1061320

Parmar, R. R., & Roy, S. (2018). MongoDB as an Efficient Graph Database: An Application of Document Oriented NOSQL Database. *Advances in Parallel Computing*, *29*, 331–358. https://doi.org/10.3233/978-1-61499-814-3-331

Pei, J., Jiang, B., Lin, X., & Yuan, Y. (2007a). *Probabilistic Skylines on Uncertain Data*.

Pei, J., Jiang, B., Lin, X., & Yuan, Y. (2007b). *Probabilistic Skylines on Uncertain Data* .

Piray, P., & Daw, N. D. (2021). Linear reinforcement learning in planning, grid fields, and cognitive control. *Nature Communications*, *12*(1). https://doi.org/10.1038/s41467-021-25123-3

Powers, D. M. W. (2020). Evaluation: from precision, recall and F-measure to ROC, informedness, markedness and correlation. *ArXiv:2010.16061 [Cs, Stat]*. https://arxiv.org/abs/2010.16061

Prakash, D. V. (2024). Deep learning for graph structured data. *Ntu.edu.sg*. https://hdl.handle.net/10356/175787

Razian, M., Fathian, M., Bahsoon, R., Toosi, A. N., & Buyya, R. (2022). Service composition in dynamic environments: A systematic review and future directions. *Journal of Systems and Software*, *188*, 111290. https://doi.org/10.1016/j.jss.2022.111290

Retzlaff, C. O., Gollob, C., Nothdurft, A., Stampfer, K., & Holzinger, A. (2024). Multi-objective optimization of cable-road layouts in smart forestry. *International Journal of Forest Engineering*, *35*(3), 444–455. https://doi.org/10.1080/14942119.2024.2380229

Robinson, I. (2015). *GRAPH DATABASES : new opportunities for connected data.* O'Reilly Media; 2nd edition .

Sahu, S., Mhedhbi, A., Salihoğlu, S., Lin, & Özsu, M. T. (2017). The ubiquity of large graphs and surprising challenges of graph processing. *Proceedings of the VLDB Endowment*, *11*(4), 420–431. https://doi.org/10.1145/3186728.3164139

Salamanos, N., Voudigari, E., & Yannakoudakis, E. J. (2017). Deterministic graph exploration for efficient graph sampling. *Social Network Analysis and Mining*, *7*(1). https://doi.org/10.1007/s13278-017-0441-6

Segaran, T., Evans, C., & Taylor, J. (2009). *Programming the Semantic Web*. "O'Reilly Media, Inc."

Shang, Z., Zgraggen, E., Buratti, B., Eichmann, P., Karimeddiny, N., Meyer, C., Runnels, W., & Kraska, T. (2021). Davos: a system for interactive data-driven decision making. *Proceedings of the VLDB Endowment*, *14*(12), 2893–2905. https://doi.org/10.14778/3476311.3476370

Shanker, S. (2024). Degree Based Search: A Novel Graph Traversal Algorithm Using Degree Based Priority Queues. *IJACSA) International Journal of Advanced Computer Science and Applications*, *15*(7), 2024. http://saiconferences.com/Downloads/Volume15No7/Paper_132-Degree_Based_Search_A_Novel_Graph_Traversal_Algorithm.pdf

Singh, B., Kumar, R., & Singh, V. P. (2021). Reinforcement learning in robotic applications: a comprehensive survey. *Artificial Intelligence Review*, *55*. https://doi.org/10.1007/s10462-021-09997-9

Singh, D., & Singh, B. (2019). Investigating the impact of data normalization on classification performance. *Applied Soft Computing*, *97*(105524), 105524. https://doi.org/10.1016/j.asoc.2019.105524

Sorrentino, G. (2022). A Skyline and ranking query odyssey: a journey from skyline and ranking queries up to f-skyline queries. *ArXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2204.04628

Srilakshmi, B., & Kumar, K. (2017). *An Efficient and Scalable Location-Aware Recommender System*.

Sucar, L. E. (2020). Probabilistic Graphical Models. In *Advances in computer vision and pattern recognition*. Springer International Publishing. https://doi.org/10.1007/978-3-030-61943-5

Suciu, D., Olteanu, D., & Koch, C. (2011). *Probabilistic Databases*. Morgan & Claypool Publishers.

Sukhwani, N., Kagita, V. R., Kumar, V., & Panda, S. K. (2021). Efficient Computation of Top-K Skyline Objects in Data Set With Uncertain Preferences. *International Journal of Data Warehousing and Mining*, *17*(3), 68–80. https://doi.org/10.4018/ijdwm.2021070104

Sunhee, & Shahabi, C. (n.d.). *Distributed Spatial Skyline Query Processing in Wireless Sensor Networks*.

Suo, S., Regalado, S., Casas, S., & Urtasun, R. (2021). TrafficSim: Learning To Simulate Realistic Multi-Agent Behaviors. *Thecvf.com*, 10400–10409. https://openaccess.thecvf.com/content/CVPR2021/html/Suo_TrafficSim_Learning_To_Simulate_Realistic_Multi-Agent_Behaviors_CVPR_2021_paper.html

Swidan, M. B., Alwan, A. A., Turaev, S., Ibrahim, H., Abualkishik, A. Z., & Gulzar, Y. (2020). Skyline Queries Computation on Crowdsourced- Enabled Incomplete Database. *IEEE Access*, *8*, 106660–106689. https://doi.org/10.1109/access.2020.3000664

Tiakas, E., Papadopoulos, A. N., & Manolopoulos, Y. (2015). Skyline queries: An introduction. *CiteSeer X (the Pennsylvania State University)*. https://doi.org/10.1109/iisa.2015.7388053

Tuunanen, T., Winter, R., & vom Brocke, J. (2024). Dealing with Complexity in Design Science Research: A Methodology Using Design Echelons: MIS Quarterly. *MIS Quarterly*, *48*(2), 427–458. https://doi.org/10.25300/MISQ/2023/16700

Versari, L., Comsa, I.-M., Conte, A., & Grossi, R. (2020). Zuckerli: A New Compressed Representation for Graphs. *IEEE Access*, *8*, 219233–219243. https://doi.org/10.1109/access.2020.3040673

Vujovic, Ž. Đ. (2021). Classification Model Evaluation Metrics. *International Journal of Advanced Computer Science and Applications*, *12*(6). https://doi.org/10.14569/ijacsa.2021.0120670

Wachi, A., & Sui, Y. (2020). Safe Reinforcement Learning in Constrained Markov Decision Processes. *PMLR*, 9797–9806. https://proceedings.mlr.press/v119/wachi20a.html

Walke, D., Micheel, D., Schallert, K., Muth, T., Broneske, D., Saake, G., & Heyer, R. (2023). The importance of graph databases and graph learning for clinical applications. *Database*, *2023*. https://doi.org/10.1093/database/baad045

Wang, Y., Li, X., Li, X., & Wang, Y. (2013). A survey of queries over uncertain data. *Knowledge and Information Systems*, *37*(3), 485–530. https://doi.org/10.1007/s10115-013-0638-6

Wang, Y., Song, B., Wang, J., Zhang, L., & Wang, L. (2016). Geometry-Based Distributed Spatial Skyline Queries in Wireless Sensor Networks. *Sensors*, *16*(4), 454. https://doi.org/10.3390/s16040454

Wang, Y., Wang, L., Li, Y., He, D., Liu, T.-Y., & Chen, W. (2013). A Theoretical Analysis of NDCG Type Ranking Measures. *Journal of Machine Learning Research*. https://doi.org/10.48550/arxiv.1304.6480

Wang, Y., Wei, W., Deng, Q., Liu, W., & Song, H. (2016). An Energy-Efficient Skyline Query for Massively Multidimensional Sensing Data. *Sensors*, *16*(1), 83. https://doi.org/10.3390/s16010083

Yacouby, R., & Axman, D. (2020, November 1). *Probabilistic Extension of Precision, Recall, and F1 Score for More Thorough Evaluation of Classification Models*. ACLWeb; Association for Computational Linguistics. https://doi.org/10.18653/v1/2020.eval4nlp-1.9

Yang, S.-Q., Yan, X., Zong, B., & Khan, A. (2012). Towards effective partition management for large graphs. *International Conference on Management of Data*. https://doi.org/10.1145/2213836.2213895

Yang, Z., Li, K., Zhou, X., Mei, J., & Gao, Y. (2018). Top k probabilistic skyline queries on uncertain data. *Neurocomputing*, *317*, 1–14. https://doi.org/10.1016/j.neucom.2018.03.052

Yin, B., Zhou, S., Zhang, S., Gu, K., & Yu, F. (2017). On Efficient Processing of Continuous Reverse Skyline Queries in Wireless Sensor Networks. *Ksii Transactions on Internet and Information Systems*, *11*(4). https://doi.org/10.3837/tiis.2017.04.006

Ying, Z., Bourgeois, D., You, J., Zitnik, M., & Leskovec, J. (2019). *GNNExplainer: Generating Explanations for Graph Neural Networks*. Neural Information Processing Systems; Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2019/hash/d80b7040b773199015de6d3b4293c8ff-Abstract.html

Yong, H., Lee, J., Kim, J., & Hwang, S. (2014). Skyline ranking for uncertain databases. *Information Sciences*, *273*, 247–262. https://doi.org/10.1016/j.ins.2014.03.044

Yuan, Y., Wang, G., Wang, H., & Chen, L. (2011). Efficient subgraph search over large uncertain graphs. *Proceedings of the VLDB Endowment*, *4*(11), 876–886. https://doi.org/10.14778/3402707.3402726

Zaman, A., Siddique, M. A., None Annisa, & Morimoto, Y. (2015). *Selecting Key Person of Social Network Using Skyline Query in MapReduce Framework*. *11*, 213–219. https://doi.org/10.1109/candar.2015.84

Zeng, Y., Chen, G., Li, K., Zhou, Y., Zhou, X., & Li, K. (2019). M-Skyline: Taking sunk cost and alternative recommendation in consideration for skyline query on uncertain data. *Knowledge-Based Systems*, *163*, 204–213. https://doi.org/10.1016/j.knosys.2018.08.024

Zhang, J., Koppel, A., Bedi, A. S., Szepesvari, C., & Wang, M. (2020). *Variational Policy Gradient Method for Reinforcement Learning with General Utilities*. Neural Information Processing Systems; Curran Associates, Inc. https://proceedings.neurips.cc/paper_files/paper/2020/hash/30ee748d38e21392de740e2f9dc686b6-Abstract.html

Zhang, K., Gao, H., Han, X., Cai, Z., & Li, J. (2020). Modeling and Computing Probabilistic Skyline on Incomplete Data. *IEEE Transactions on Knowledge and Data Engineering*, *32*(7), 1405–1418. https://doi.org/10.1109/tkde.2019.2904967

Zhang, S., Ray, S., Lu, R., Zheng, Y., Guan, Y., & Shao, J. (2021). Achieving Efficient and Privacy-Preserving Dynamic Skyline Query in Online Medical Diagnosis. *IEEE Internet of Things Journal*, 1–1. https://doi.org/10.1109/jiot.2021.3117933

Zhang, W., Lin, X., Zhang, Y., Muhammad Aamir Cheema, & Zhang, Q. (2012). Stochastic skylines. *ACM Transactions on Database Systems*, *37*(2), 1–34. https://doi.org/10.1145/2188349.2188356

Zhang, X., & Liu, C.-A. (2023). Model averaging prediction by K-fold cross-validation. *Journal of Econometrics*, *235*(1), 280–301. https://doi.org/10.1016/j.jeconom.2022.04.007

Zhang, Y., Zhang, W., Lin, X., Jiang, B., & Pei, J. (2011). Ranking uncertain sky: The probabilistic top-k skyline operator. *Information Systems*, *36*(5), 898–915. https://doi.org/10.1016/j.is.2011.03.008

Zhang, Z., Hua, Beng, L., Ooi, C., Tung, A., Lu, H., & Ooi, B. (n.d.). *Generic Analysis and Methods for Computing Skyline Variants*.

Zhang, Z., Lu, H., Ooi, B. C., & Tung, A. K. H. (2009). Understanding the meaning of a shifted sky: a general framework on extending skyline query. *The VLDB Journal*, *19*(2), 181–201. https://doi.org/10.1007/s00778-009-0148-z

Zheng, K., Yang, R.-J., Xu, H., & Hu, J. (2016). A new distribution metric for comparing Pareto optimal solutions. *Structural and Multidisciplinary Optimization*, *55*(1), 53–62. https://doi.org/10.1007/s00158-016-1469-3

Zhou, X., Li, K., Zhou, Y., & Li, K. (2016). Adaptive Processing for Distributed Skyline Queries over Uncertain Data. *IEEE Transactions on Knowledge and Data Engineering*, *28*(2), 371–384. https://doi.org/10.1109/tkde.2015.2475764