Event Graph Optimization in RFI Text Documents using Hierarchical Reinforcement Learning and Human Feedback

Sareh Ahmadinasab

A Thesis

in

The Department

of

Concordia Institute for Information System Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Quality System Engineering) at

Concordia University

Montréal, Québec, Canada

April 2025

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: Sareh Ahmadinasab

Entitled: Event Graph Optimization in RFI Text Documents using Hierarchical Reinforcement Learning and Human Feedback

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Quality System Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

______________________________ Chair
Dr. Yong Zeng

______________________________ Examiner
Dr. Yong Zeng

______________________________ Examiner
Dr. Hua Ge

______________________________ Supervisor
Dr. Jun Yan

Approved by ______________________________
Dr. Chun Wang, Chair
Department of Concordia Institute for Information System Engineering

__________ 2025

____
Dr. Mourad Debbabi, Dean
Gina Cody School of Engineering and Computer Science

Abstract

Event Graph Optimization in RFI Text Documents using Hierarchical Reinforcement Learning and Human Feedback

Sareh Ahmadinasab

Requests for Information (RFIs) are essential tools for facilitating communication among stakeholders in construction project management. They serve as formal inquiries to clarify ambiguities, resolve uncertainties, and enhance coordination between project teams, ultimately supporting effective project execution. RFIs play a critical role in maintaining workflow efficiency, reducing misinterpretations, and addressing unforeseen challenges that arise during the construction process. However, despite their importance, RFIs can sometimes highlight underlying systemic inefficiencies or anomalies that, if left unaddressed, contribute to cost overruns, schedule delays, and project quality issues. Identifying the root causes of such anomalies within RFIs is crucial for mitigating risks and improving decision-making. Root Cause Analysis (RCA) is commonly employed to trace these issues back to their sources, allowing project managers and engineers to implement corrective measures. However, existing RCA approaches, many of which are based on event graphs, tend to rely on predefined rules or statistical correlations, which may not fully capture the dynamic and evolving nature of construction-related issues. To address these challenges, this study proposes a novel approach that integrates human feedback-augmented reinforcement learning to enhance RCA for the event graphs of text-based RFIs. Our method leverages expert insights as a core component of the HRL loop in optimizing and improving the accuracy and reliability of causal and temporal graphs by effectively identifying and correcting errors within the graphs themselves. Importantly, our method is explicitly designed for structured RFI text data in which events have been manually pre-extracted as part of a preprocessing pipeline. Specifically, we employ hierarchical reinforcement learning (HRL) to systematically decompose the problem into multiple levels of decision-making, allowing for more structured learning and adaptation. To validate our approach, we conduct experiments using the Causal-TimeBank dataset, a benchmark corpus annotated with explicit temporal and causal relationships. The reason for choosing this benchmark is largely due to the lack of publicly available RFIs with enriched text data and annotated causal/temporal events. Experimental results demonstrate that our method outperforms conventional RCA techniques by effectively identifying and correcting errors within causal and temporal graphs. The integration of human expertise ensures that the model remains adaptable to real-world complexities, enhancing its ability to capture nuanced relationships that might otherwise be overlooked by automated approaches. Ultimately, this work contributes to the advancement of intelligent RCA systems by combining human intuition with machine learning to create more robust, interpretable, and actionable root cause analyses in construction project management.

Keywords: Event Graphs, Hierarchical Reinforcement Learning, Human Feedback, Request for Information, Root Cause Analysis

Acknowledgments

I would like to sincerely thank my supervisor, Prof. Jun Yan, for his invaluable guidance, continuous support, and encouragement throughout the course of this research.

I would like to acknowledge the support from the Natural Sciences and Engineering Research Council of Canada (NSERC) and Concordia University. Their generous support made it possible to explore and advance research. I would also like to extend my appreciation to the members of the Concordia Institute for Information Systems Engineering (CIISE) for providing a collaborative and inspiring research environment.

Lastly, heartfelt thanks to my family and friends for their patience, encouragement, and unwavering support throughout my graduate studies.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Chapter 1 provides a brief introduction to the background, basic concepts, motivation, and organization of this research. Finally, the outline of this thesis is given. A comprehensive literature review on event graph optimization for the request for information (RFI) with hierarchical reinforcement learning and human feedback is presented in Chapter 2.

1.1 Context

Construction projects involve multiple stakeholders, including contractors, engineers, and project managers, who rely on structured communication tools to address uncertainties and resolve issues. Among these tools, Requests for Information (RFIs) serve as formal queries submitted during a project's lifecycle to clarify ambiguities, missing information, or conflicting instructions in design and contract documents (Hanna et al., 2012). RFIs help ensure project accuracy and efficiency, but they can also indicate underlying inefficiencies within project workflows.

RFIs are a critical communication tool between the design team and the construction team, facilitating the exchange of information needed to clarify ambiguities, address missing details, and resolve conflicts within project documentation (Bhat, 2017). Despite their significance in ensuring project alignment and mitigating risks, RFIs are often viewed negatively due to their time-consuming and labor-intensive nature (Aibinu et al., 2020).

One of the primary reasons for this negative perception is the effort required to initiate, review, and respond to RFIs. The process typically involves multiple stakeholders, including contractors, designers, consultants, and project managers, all of whom must analyze the inquiry, provide necessary clarifications, and ensure that the response aligns with contractual obligations and project requirements (Afzal et al., 2023). This multi-step process often leads to delays and inefficiencies, especially when RFIs accumulate in high volumes on large-scale projects (Afzal et al., 2024).

Traditional methods of handling RFIs rely on manual processes and expert judgment, which are time-consuming, error-prone, and inefficient, particularly for large-scale projects (Love et al., 2014). The response time for RFIs is critical, as prolonged delays can disrupt construction schedules, create bottlenecks, and increase project costs. If RFIs are not addressed in a timely manner, construction teams may be forced to proceed with work based on assumptions or wait for clarifications, leading to schedule slippage and resource misallocation (Kelly and Ilozor, 2020; Panahi et al., 2023; Shim et al., 2016). Studies have shown that prolonged response times are a leading cause of inefficiencies in project workflows, with certain RFIs remaining unresolved for weeks or even months (Özoğul and Ergen, 2024).

Additionally, delays or lack of responses to RFIs can result in frustration, disputes, and mistrust among project team members (Afzal et al., 2023; Philips-Ryder et al., 2013). When RFIs are repeatedly ignored or inadequately addressed, contractors and subcontractors may perceive a lack of collaboration and accountability from the design or project management team (Shrestha et al., 2023). This can lead to escalations, legal claims, and contractual disputes, further straining relationships and increasing administrative overhead (Alrasheed et al., 2023; Zhang et al., 2020).

Furthermore, RFIs are often not prioritized effectively, as project teams may struggle with sorting, tracking, and managing numerous RFIs at different project phases (Shim et al., 2016). The traditional manual methods of logging, reviewing, and responding to RFIs make it difficult to ensure transparency and efficiency in decision-making. This has led to increasing efforts to automate RFI workflows using machine learning (ML) models, which aim to classify, prioritize, and even auto-generate responses to RFIs (Afzal et al., 2024; Wang et al., 2023).

To improve the efficiency of analyzing RFIs, advanced techniques such as causal and temporal modeling can provide deeper insights into project risks and systematic issues. This study introduces a Hierarchical Reinforcement Learning (HRL) approach augmented with Human Feedback (HF) to refine causal and temporal graphs, improving the accuracy of Root Cause Analysis (RCA) in RFIs (Wang et al., 2023). By leveraging human expertise within the RL loop, the proposed method enhances RCA in construction project management, reducing inefficiencies and improving decision-making processes. While this study focuses on event graph optimization from structured RFI text, the proposed method aligns with digital construction workflows. In practice, RFIs are commonly managed through structured project platforms such as Aconex or Procore, where textual communication is stored in a digital format. Our approach assumes that event data are available in digital form. The initial event graphs can then be constructed manually or semi-automatically and passed to the HRLHF model for optimization.

1.2 Motivation and Background

Construction projects often face delays, cost overruns, and quality issues due to uncertainties that emerge during execution. RFIs are crucial for identifying and addressing these issues early; however, manually analyzing thousands of RFIs across multiple projects is impractical (Aibinu et al., 2020).

Several existing approaches have attempted to automate the processing of RFIs, including: Natural Language Processing (NLP) to extract structured information from RFIs (Afzal et al., 2023), Machine Learning (ML) techniques for classifying and prioritizing RFIs (Gopalakrishnan et al., 2023), Graph-based models for representing relationships between project entities (Li et al., 2023).

While these approaches contribute to improved RFI management, they often fall short in capturing the underlying causal and temporal dependencies between events. Most methods lack interpretability and adaptability, limiting their effectiveness in dynamic project environments (Liu et al., 2020).

Reinforcement Learning (RL) has shown promise in various domains for learning optimal decision-making strategies. However, RL models often struggle with real-world complexity and require extensive training data (Sutton and Barto, 2018). This challenge is addressed by incorporating Hierarchical RL (HRL), which breaks down decision-making into structured levels, and Human Feedback (HF), which ensures model corrections based on domain expertise (Bai et al., 2024).

The motivation behind this study is to develop a scalable and interpretable approach that integrates Hierarchical Reinforcement Learning (HRL) and Human Feedback (HF) to improve Root Cause Analysis (RCA) in construction projects. Specifically, this research focuses on optimizing causal and temporal graphs derived from data, which serve as critical tools for understanding project delays, inefficiencies, and communication breakdowns.

This approach will offer significant benefits to both researchers and industry stakeholders. The optimized causal and temporal graphs will provide a structured and data-driven representation of project events, helping researchers to better analyze patterns, predict risks, and develop models that improve decision-making. By enhancing the accuracy and interpretability of these graphs, researchers will gain deeper insights into how different factors contribute to delays and cost overruns in construction projects.

For industry professionals, including project managers, engineers, and contractors, the optimized graphs will enable more effective RCA by identifying key dependencies between events and their impacts on the project timeline. This can lead to faster resolution of RFIs, improved project coordination, and proactive risk management strategies. Additionally, the integration of HRL and HF will allow continuous learning and adaptation based on real-world feedback, making the system more robust and applicable across diverse construction scenarios. Ultimately, this study aims to bridge the gap between academic research and industry needs, providing a scalable solution for improving efficiency, reducing disputes, and enhancing project outcomes in the construction sector.

1.3 Objectives and Contributions

The primary objective of this thesis is to develop an advanced reinforcement learning (RL) framework tailored for Root Cause Analysis (RCA) in large-scale construction projects. Traditional RCA methods struggle with handling the complexity and semi-structured nature of Request for Information (RFI) data, often resulting in suboptimal issue resolution and delays. This research addresses these challenges by integrating two key advancements into reinforcement learning:

- **Hierarchical Reinforcement Learning (HRL)**

  By structuring the decision-making process into hierarchical levels, HRL allows the model to learn at different levels of abstraction and systematically optimize causal and temporal relationships within RFI data. This multi-level learning strategy enables the RL agent to decompose complex problem-solving tasks, enhancing its ability to handle dynamic project environments.

- **Human Feedback (HF)**

A key limitation of RL in real-world applications is the potential for incorrect generalizations. To mitigate this, expert knowledge is integrated into the learning loop as a feedback mechanism. This ensures that model predictions align with domain expertise and real-world constraints, improving both interpretability and performance. The feedback mechanism enables continuous refinement of causal and temporal graphs, ensuring they remain accurate and contextually relevant (Bai et al., 2024).

By optimizing causal and temporal graphs derived from RFI data, the proposed method aims to enhance the accuracy of anomaly detection, streamline issue resolution, and ultimately reduce delays and cost overruns in large-scale construction projects.

This study aims to develop a scalable and interpretable approach that combines HRL and HF to improve Root Cause Analysis (RCA) in construction project management. The key contributions of this work include:

- **Application of HRL with HF for RFI Analysis:**
  We use HRL with HF to iteratively improve causal and temporal graphs for RFIs, enabling the model to learn from expert insights and enhance its accuracy in capturing complex causal and temporal relationships in RFI data.

- **Optimization of Causal and Temporal Graphs Using an MDP-Based HRL Framework:**
  The HRL approach is structured through an MDP framework that separates RFI causal and temporal analysis into high-level structure adjustments and low-level link refinements, enabling more accurate graph tuning for better root cause analysis.

- **Implementation of Iterative Learning for Continuous Improvement:**
  Our iterative learning approach integrates HF in each cycle to adjust the model's reward function, resulting in significant improvements in evaluation metrics including Precision, Recall, and F1 Score.

- **Development of a Dedicated Training Platform for Structured Evaluation:** We created a training platform for evaluating the HRL-based causal and temporal graph model across various datasets, providing a standardized testing environment that supports detailed performance analysis and future research in RCA for RFIs.

By leveraging HRL and HF, this thesis presents a novel reinforcement learning framework that enhances the accuracy of RCA in RFIs. The proposed approach aims to reduce project delays and cost overruns by improving issue identification, causal reasoning, and corrective decision-making in construction project management. A research paper titled "Event Graph Optimization for Request for Information (RFI) with Hierarchical Reinforcement Learning and Human Feedback" has been submitted to the conference (Joint CSCE Construction Specialty Conference / ASCE Construction Research Congress (CRC)) in February 2025.

1.4 Thesis Outline

The thesis is structured as follows:

- **Chapter 1:** "Introduction" provides an overview of the research background, motivation, and significance of RFIs in construction project management. The chapter also outlines the objectives and contributions of this research.

- **Chapter 2:** "Literature Review" explores existing research on RFIs, Root Cause Analysis, causal and temporal graph modeling, and the role of Hierarchical Reinforcement Learning in construction data analysis and integration of human feedback in machine learning.

- **Chapter 3:** "Methodology" details the proposed HRL framework, including its

Markov Decision Process formulation, hierarchical agent structure, and human feedback integration, along with modeling the reward to optimize causal and temporal graphs and experimental setup to train and test the RL-based approach.

- **Chapter 4:** “Experiments and Results” presents the datasets, evaluation metrics, and experimental results that demonstrate the effectiveness of the proposed HRL-based RCA method.
- **Chapter 5:** “Discussion” interprets the findings, identifies limitations and challenges, and potential applications and future research directions.
- **Chapter 6:** “Conclusion and Future Work” summarizes the key findings of this research and outlines potential advancements in HRL-based RCA for construction data analysis.

Chapter 2

Literature Review

2.1 Request for Information (RFIs)

A Request for Information (RFI) is a formal document used in construction project management to seek clarification regarding project specifications, contract terms, or design details (Hanna et al., 2012). RFIs serve as a critical communication tool that helps project teams identify and resolve ambiguities, ensuring that construction progresses smoothly without unnecessary conflicts or costly errors (Love et al., 2014). They are particularly important for quality assurance and compliance, allowing for early detection of potential nonconformities and enabling corrective measures to be taken before they escalate into significant project delays (Aibinu et al., 2020).

2.1.1 Types of RFIs and Their Impact

RFIs play a crucial role in maintaining workflow efficiency and ensuring that construction activities proceed smoothly. Studies have shown that unresolved RFIs can lead to significant delays, increased costs, and disputes among stakeholders (Love et al., 2014;

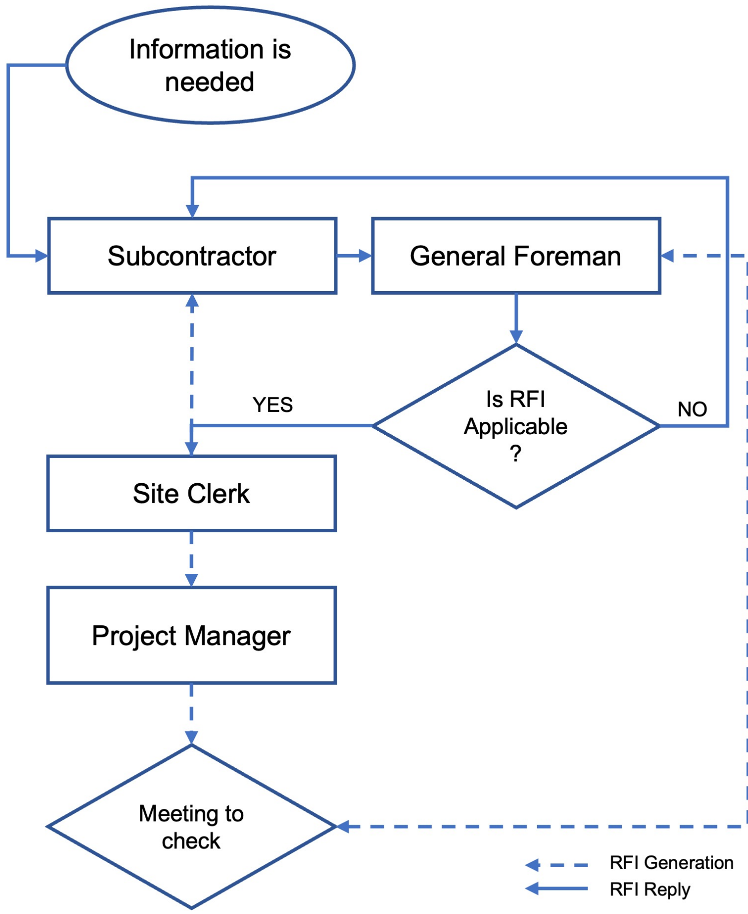

Figure 2.1: Framework of the RFI in projects. (Mohamed et al., 1999)

Shim et al., 2016). RFIs help address ambiguities in design drawings, technical specifications, and contractual agreements (Hughes et al., 2013). By proactively resolving uncertainties, RFIs reduce the likelihood of costly rework and legal disputes (Papajohn and El Asmar, 2021) and facilitate structured communication among project teams, promoting transparency and accountability Afzal et al. (2023). RFIs can be categorized based on their purpose and impact. Common RFI categories include:

- Design Clarification RFIs: When inconsistencies exist between project drawings, requiring engineers and architects to provide additional detailsLove2014 (Love et al., 2014).

- Coordination RFIs: Address conflicts between different project disciplines, such as mechanical and electrical systems, to avoid on-site clashes (Morales et al., 2022).

- Specification RFIs: Requests for additional details about material specifications, building codes, or construction methods (Love et al., 2014).
- Contractual RFIs: Clarifications regarding project scope, responsibilities, or terms of agreements between stakeholders (Hughes et al., 2013).

Unresolved RFIs can lead to significant delays and cost increases, especially when they affect critical-path tasks (Papajohn and El Asmar, 2021). For instance, an unresolved RFI concerning HVAC ductwork and electrical conduit installation can halt multiple construction activities, increasing labor costs and pushing project deadlines (Shim et al., 2016).

2.1.2 The Structure of an RFI

As shown in Fig. 2.2 an RFI typically consists of the following key components (Labs, 2024):

- **Title and Identification Number**: Each RFI is assigned a unique title and reference number for tracking and record-keeping purposes.
- **Date of Submission**: The official date when the RFI is submitted, ensuring a clear timeline for processing and response.
- **Project Information**: Essential details about the project, including its name, location, and contract number, to provide context for the inquiry.
- **Requesting Party**: The individual or entity submitting the RFI, such as a contractor, subcontractor, or consultant, who requires clarification or information.
- **Recipient Information**: The designated recipient responsible for addressing the RFI, which may include an architect, engineer, or project manager.

- **Subject and Description**: A concise summary outlining the issue, ambiguity, or missing information prompting the request.

- **Supporting Documents**: Attachments such as drawings, specifications, contracts, or previous correspondence that provide additional context for the inquiry.

- **Proposed Resolution or Inquiry**: A clearly formulated question or suggested resolution to expedite the response process.

- **Response Section**: The section where the recipient provides clarifications, instructions, or additional information to resolve the RFI.

- **Deadline for Response**: The required timeframe within which a response must be provided to prevent project delays and ensure timely decision-making.

The structured format of an RFI enhances transparency and accountability, helping project stakeholders manage uncertainties efficiently and maintain workflow continuity.

2.1.3 Application, Automation and AI in RFI Processing

Recent advances in Natural Language Processing (NLP) and Artificial Intelligence (AI) have introduced innovative solutions to automate and streamline RFI analysis (Afzal et al., 2023; Jallan et al., 2019). Text mining and machine learning techniques enable automated classification of RFIs, extraction of key topics, and detection of recurring issues, improving project teams' ability to respond efficiently (Yilmaz and Ergen, 2024).

Key innovations in RFI automation include:

- Topic Modeling and Text Clustering: Techniques like Latent Dirichlet Allocation (LDA) and K-Means Clustering help categorize RFIs into meaningful groups, revealing patterns that aid in proactive issue resolution (Yilmaz and Ergen, 2024).

RFI NAME		RFI ID	
REQUESTING PARTY		RESPONDING PARTY	
RFI SUBMISSION DEADLINE		DATE OF RFI RESPONSE	

ORGANIZATION NAME		CONTACT NAME	
ADDRESS		CONTACT TITLE	
		PHONE	
		EMAIL	
		WEBSITE	

REQUEST DESCRIPTION

RESPONSE

RESPONSE PREPARED BY			
ATTACHMENTS?		NUMBER OF PAGES	

Figure 2.2: Request for Information in Construction project Templates (Smartsheet, 2025)

- Semantic Annotation and Named Entity Recognition (NER): AI-driven entity recognition models identify project-specific terms, hazards, and requirements from RFIs, enhancing decision support systems (Thompson et al., 2020).

- RFI Recommender Systems: AI-powered recommender systems assist project teams by identifying similar past RFIs and suggesting pre-existing solutions, significantly reducing response times (Panahi et al., 2023).

Building Information Modeling (BIM) has emerged as a powerful tool for reducing the volume and impact of RFIs. By providing a centralized digital model that integrates all project disciplines, BIM helps stakeholders visualize design conflicts before construction begins, thereby minimizing the need for RFIs (Morales et al., 2022). BIM enables early detection of conflicts between structural, mechanical, and electrical components, reducing the number of coordination RFIs. By integrating project documentation within BIM platforms, AI tools can automatically generate preliminary responses to common RFI queries. BIM-based RFI tracking systems help identify recurring issues, enabling data-driven decision-making for future projects.

Table 2.1: Overview of RFI-related Research Papers

Paper	**Key Focus**	**Methodology**	**Findings**
Papajohn and El Asmar (2021)	Analyzing how alternative delivery methods impact RFI response times	Empirical analysis of RFI datasets	Alternative delivery methods reduce RFI response times significantly
Afzal et al. (2023)	Using text mining and NLP techniques to extract key insights from RFIs	Natural Language Processing (NLP) & Text Mining	Text mining can enhance RFI classification and issue detection
Morales et al. (2022)	Integrating BIM for proactive conflict detection to reduce RFIs	Building Information Modeling (BIM)	BIM reduces coordination RFIs and improves project planning
Özoğul and Ergen (2024)	Employing NLP and ML models for metadata extraction from RFIs	Machine Learning (ML) & NLP	ML-based metadata extraction improves RFI documentation efficiency
Shim et al. (2016)	Case study exploring RFI management best practices	Qualitative case study analysis	Structured RFI management can prevent project delays
Yilmaz and Ergen (2024)	Comparative study of clustering methods for RFI analysis	Unsupervised clustering & visualization	Clustering methods reveal patterns in RFI submission trends

2.1.4 Future Directions and Challenges

Despite these advancements, several challenges remain in RFI automation:

- Handling Unstructured Text: RFIs are often unstructured and written in diverse formats, requiring sophisticated NLP models to extract actionable insights.

- Ensuring Domain-Specific Accuracy: Construction RFIs contain technical language, abbreviations, and project-specific terminology, making it difficult for generic NLP models to interpret them accurately.
- Integrating AI with Existing Workflows: While AI-driven RFI management shows promise, adoption barriers exist due to resistance from industry professionals who are accustomed to traditional RFI handling processes.

As research in NLP, AI, and BIM integration progresses, the future of RFI management is expected to move towards fully automated, predictive analytics-driven systems that not only respond to RFIs but also prevent their occurrence in the first place.

2.2 Importance of Root Cause Analysis for RFIs

Root Cause Analysis (RCA) is a structured approach used to identify the fundamental causes of project issues documented within Request for Information (RFI) processes. Given the complexity of modern construction projects, RFIs play a critical role in project communication, helping teams resolve ambiguities and address technical uncertainties (Liu et al., 2020). However, without an effective RCA framework, RFIs can become a bottleneck, leading to cost overruns, delays, and inefficiencies (Gopalakrishnan et al., 2023).

2.2.1 Challenges in Traditional RCA Approaches

Traditional RCA in RFIs relies on manual assessments and expert judgments, making the process labor-intensive, subjective, and prone to biases (Afzal et al., 2023). Conventional methods often focus on surface-level causes rather than deep structural inefficiencies, leading to recurring issues in construction projects. These methods typically involve:

- Reviewing project documents manually, which is time-consuming.

- Expert-driven decision-making, which can introduce inconsistencies.
- Delayed issue resolution, impacting the overall project timeline.

2.2.2 The Need for Data-Driven RCA in RFIs

Recent advances in machine learning (ML), natural language processing (NLP), and causal inference offer promising solutions for automating and enhancing RCA in RFIs. Research has demonstrated that applying unsupervised learning techniques such as Latent Dirichlet Allocation (LDA) and K-means clustering can significantly improve the identification of root causes within large RFI datasets (Yilmaz and Ergen, 2024). These techniques enable:

- Automated pattern recognition in RFI documentation.
- Detection of structural discrepancies and design conflicts before they escalate.
- Improved response efficiency through predictive analytics.

Incorporating causal inference models allows for systematic RCA, linking RFI queries to specific project inefficiencies and anomalies (Li et al., 2023). By establishing causal and temporal relationships, ML-driven RCA can predict future RFIs and recommend preemptive actions, reducing project disruptions.

2.2.3 Enhancing RCA with AI and RL in Construction Projects

The integration of Artificial Intelligence (AI), particularly Reinforcement Learning (RL), with Root Cause Analysis (RCA) has demonstrated significant improvements in managing Requests for Information (RFIs) and reducing project inefficiencies. AI-driven RCA leverages machine learning models to identify, analyze, and mitigate issues proactively, reducing response times and minimizing costly project delays. Reinforcement Learning, a subset of AI, enhances RCA by enabling autonomous systems to iteratively learn from project data, optimizing decision-making over time (Panahi et al., 2023).

Some of the key ways AI and RL enhance RCA in construction projects include:

- **Automated Causal Discovery**: AI-powered RCA frameworks use reinforcement learning to analyze large datasets and establish causal relationships between project variables, uncovering hidden dependencies in construction workflows.
- **Adaptive Learning for RCA**: RL-based models continuously refine their understanding of project anomalies, dynamically adjusting their strategies based on real-time construction data and human feedback (Wang et al., 2023).
- **AI-Driven Predictive Analytics**: Machine learning techniques predict potential RFIs by analyzing historical project data, allowing for proactive risk mitigation and issue resolution (Morales et al., 2022).
- **Intelligent Conflict Resolution**: AI-enhanced RCA frameworks automate clash detection and coordination across different project disciplines, minimizing costly rework and improving project execution (Zhang et al., 2024).

Studies have shown that AI-enhanced RCA frameworks can reduce RFI resolution times by up to 40%, significantly improving project efficiency and reducing disputes (Panahi et al., 2023).

As construction projects grow in complexity, the adoption of AI and reinforcement learning in RCA will be essential for ensuring data-driven decision-making, improving issue resolution accuracy, and mitigating risks before they escalate. AI-powered RCA provides a scalable and adaptive approach that aligns with the evolving needs of modern construction management, ultimately leading to increased project success rates and cost-effectiveness.

2.3 Causal and Temporal Graphs in Construction Data Analysis

Causal and temporal graphs provide a structured representation of event relationships, allowing project managers to understand dependencies and predict potential delays (Love et al., 2014). These graphs model interactions between various project factors, such as material delivery schedules, regulatory compliance, and contractor performance. By leveraging causal and temporal graph analysis, decision-makers can proactively address root causes of project disruptions (Liu et al., 2020).

Traditional project management tools often struggle to capture the dynamic and complex interdependencies inherent in construction projects(Bai et al., 2024). To address this challenge, recent studies have introduced advanced methodologies:

- **Causal Temporal Graph Convolutional Neural Networks (CTGCN)**: This novel, scalable method deduces causal relationships in large observational data and integrates them into a Temporal Graph Convolutional Network (TGCN) architecture. This approach overcomes limitations of requiring a priori domain knowledge, facilitating self-adaptation and scalability in diverse large-scale applications (Langbridge et al., 2023).

- **Long-Term Prediction on Graph Data with Causal Network**: This research addresses the challenge of accurate and stable long-term predictions in complex spatiotemporal systems by leveraging causal network structures (Liu et al., 2024).

Despite these advancements, existing methods may still fall short in capturing intricate dependencies, underscoring the need for more robust learning techniques such as reinforcement learning (RL). Reinforcement learning enables systems to autonomously learn optimal behaviors through trial-and-error interactions within their environment. In the construction industry, RL has been applied to various domains:

- **Building Energy Management**: RL algorithms have been utilized to optimize energy consumption in buildings, leading to more efficient and sustainable operations (Asghari et al., 2022).
- **Infrastructure Management**: RL methods have been employed to predict construction price indices, safety indicators, and building lifespan, enhancing the decision-making process in infrastructure projects (Asghari et al., 2022).
- **Construction Machinery Operation**: RL facilitates the development of autonomous construction machinery capable of adapting to dynamic site conditions, thereby improving operational efficiency (Xu and Garc de Soto, 2022).

Moreover, the integration of RL with Internet of Things (IoT) technologies has led to the development of autonomous resource management systems in construction. These systems utilize deep reinforcement learning to process real-time data from IoT sensors, enabling dynamic and efficient resource allocation across multiple projects (Soleymani et al., 2022).

In summary, the adoption of causal and temporal graph analyses, augmented by advanced methodologies like reinforcement learning, holds significant promise for addressing the complexities of modern construction projects. These approaches provide a comprehensive framework for understanding project dynamics, predicting potential issues, and implementing proactive measures to mitigate disruptions, thereby enhancing the efficiency and success rate of construction endeavors.

2.4 Overview of Hierarchical Reinforcement Learning

Hierarchical Reinforcement Learning (HRL) is a machine learning approach that decomposes complex decision-making processes into hierarchical levels, facilitating efficient learning and adaptation, especially in environments with structured dependencies like construction project management. By introducing higher-level decision policies that guide lower-level actions, HRL reduces computational complexity and enhances interpretability (Sutton and Barto, 2018). This enables efficient learning and adaptation, particularly in environments with structured dependencies, such as construction project management. HRL introduces higher-level decision policies that guide lower-level actions, reducing computational complexity and improving interpretability (Wang et al., 2023).

Recent advancements have further refined HRL methodologies. The Causality-Driven Hierarchical Reinforcement Learning (CDHRL) framework, for instance, autonomously constructs hierarchical structures by leveraging causal relationships among environmental variables. This causality-driven approach enhances exploration efficiency and is particularly beneficial in complex environments (Peng et al., 2022).

In the construction industry, the integration of HRL has shown promise in optimizing various processes. A comprehensive review highlights the application of reinforcement learning methods in domains such as building energy management, infrastructure maintenance, and construction machinery operation. These applications have led to improved decision-making and operational efficiency (Asghari et al., 2022).

Moreover, HRL has been employed in temporal pattern prediction tasks, such as forecasting stock prices and vehicle steering angles. By combining deep learning with HRL, researchers have achieved significant improvements in training speed, stability, and prediction accuracy over standard reinforcement learning approaches (Johnson and Dana, 2023).

The application of HRL in construction data analysis offers opportunities to enhance Request for Information (RFI)-based Root Cause Analysis (RCA). By iteratively refining causal and temporal graphs, HRL can optimize the identification and resolution of underlying issues, leading to more efficient project management and reduced delays.

2.5 Integration of Human Feedback in Machine Learning

The integration of human expertise into machine learning models is essential for enhancing decision-making in complex domains. Human feedback ensures that learning systems align with domain-specific knowledge, thereby reducing errors and improving model interpretability (Bai et al., 2024).

2.5.1 Interactive Learning

Interactive learning methods enable machine learning agents to engage directly with human experts or users, allowing for continuous refinement and adaptation by soliciting expert advice, clarifications, or real-time feedback while learning as shown in Figure 2.3. This dynamic interaction enhances generalization and improves decision-making accuracy by incorporating human insights into the learning process(Bignold et al., 2021). This combination of interactive learning and human feedback is particularly valuable in domains such as construction Root Cause Analysis (RCA), where expert validation is crucial for accurately identifying causal relationships and project dependencies. Unlike purely data-driven methods, interactive models leverage domain expertise to improve reliability and interpretability (Gopalakrishnan et al., 2023). By iteratively refining causal and temporal graph structures with human input, these systems adapt to complex construction scenarios with greater precision, ensuring decisions are informed by both empirical data and expert knowledge (Mosqueira-Rey et al., 2023).

2.5.2 Imitation Learning

Imitation learning, also known as learning from demonstrations, is a technique where an agent acquires a policy by replicating expert behavior fig 2.4. Instead of learning through

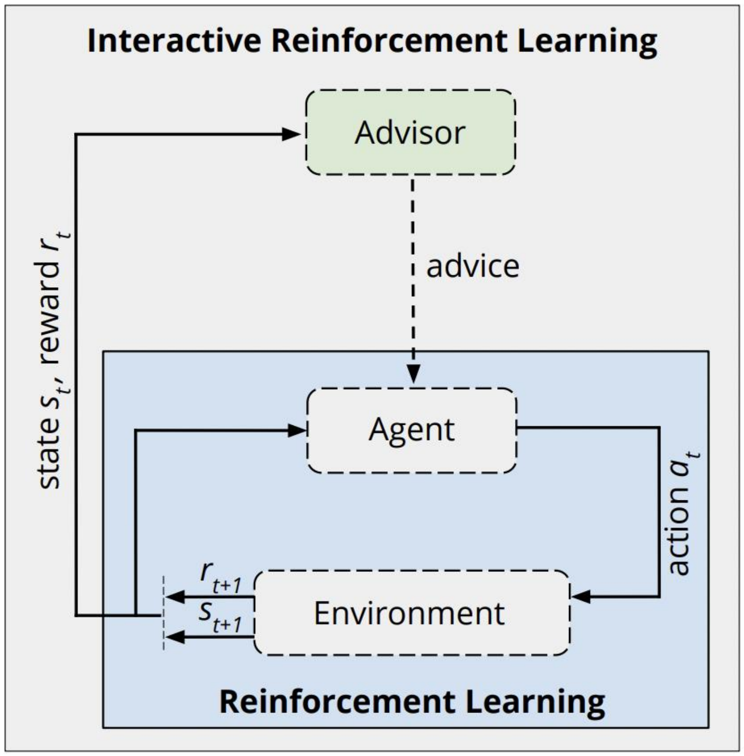

Figure 2.3: Interactive reinforcement learning approach. An illustration showing the involvement of an advisor and its relation to the traditional reinforcement learning process.

trial and error, the agent observes expert-provided sample trajectories or actions and mimics the demonstrated behavior to develop optimal decision-making patterns. This approach is particularly useful in scenarios where manually defining a reward function is complex or infeasible. By leveraging imitation learning, machine learning models can accelerate training, improve sample efficiency, and enhance decision-making in fields such as autonomous systems, robotics, and construction Root Cause Analysis (RCA) (Zuo et al., 2017).

2.5.3 Human-in-the-Loop Reinforcement Learning

Human-in-the-loop reinforcement learning (HITL RL) incorporates expert supervision into the reinforcement learning training process, ensuring that learned policies remain aligned with real-world constraints. This approach enhances model robustness and accelerates convergence by guiding agents toward optimal solutions. In dynamic environments,

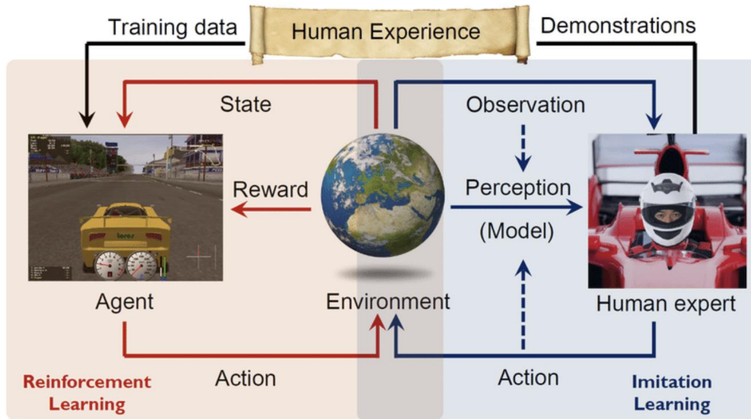

Figure 2.4: The framework of RL, imitation learning, and their integration and use of human experts to pre-train a model.

such as construction project management, HITL RL allows for continuous adaptation to changing conditions, leveraging human insights to navigate complex decision-making landscapes effectively (Retzlaff et al., 2024).

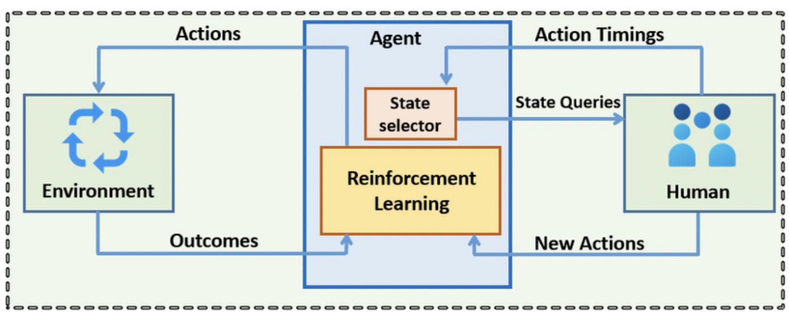

Figure 2.5: Human-in-the-loop reinforcement learning framework.

Unlike HITL RL, which primarily integrates human feedback in real time to adjust an agent's learning trajectory through corrections, demonstrations, or direct reward shaping and focuses on refining policy learning in a flat RL setting, HRLHF introduces a structured, multi-level decision-making approach, which enables a more scalable approach to complex, structured tasks like event graph optimization (Wang et al., 2023). It decomposes the problem into distinct high-level and low-level agent roles. In HRLHF, human feedback is leveraged iteratively, shaping reward functions and policy refinements at different decision levels rather than directly intervening at every step. This distinction makes

HRLHF particularly effective in domains where decision-making involves multiple levels of abstraction, such as RFI analysis in construction, where structured relationships between events must be learned and optimized.

2.5.4 Applications in Construction

In the construction industry, the integration of human feedback into machine learning models has shown significant promise. For instance, human-in-the-loop approaches have been employed to enhance the performance of construction robotics, where human oversight ensures that automated systems operate safely and effectively alongside human workers. This collaboration not only improves the efficiency of construction processes but also fosters trust in automated systems among the workforce (Shayesteh and Jebelli, 2022).

Moreover, incorporating human feedback in predictive models aids in accurately identifying potential project delays and cost overruns. Experts can provide contextual information that pure data analytics might miss, leading to more reliable forecasts and better-informed decision-making. This human-machine collaboration is pivotal in managing the complexities inherent in construction projects.

2.6 Hierarchical Reinforcement Learning (HRL) and Human Feedback in RCA

Integrating Hierarchical Reinforcement Learning (HRL) with human feedback significantly enhances Root Cause Analysis (RCA) in Request for Information (RFI) processes by combining structured learning with expert-guided refinements. This approach leverages HRL to iteratively improve causal and temporal graphs, incorporating domain expertise to correct errors and optimize decision-making (Bai et al., 2024).

Recent advancements in HRL have introduced frameworks that incorporate expert feedback to refine decision-making in complex, high-dimensional environments. Studies have shown that reinforcement learning models incorporating expert demonstrations outperform standard RL models in structured decision-making tasks (Zhou et al., 2025). Human feedback plays a critical role in guiding hierarchical policies, reducing exploration inefficiencies, and improving the model's adaptability to domain-specific constraints (Zhang et al., 2024).

In construction project management, HRL and human feedback integration have been applied to enhance dynamic scheduling, resource allocation, and risk mitigation. For instance, reinforcement learning methods have been utilized to optimize construction logistics by learning from past project data and expert interventions. The ability to adjust reinforcement learning policies in real time based on expert inputs ensures that decisions remain interpretable and aligned with industry best practices(Wang et al., 2023).

Moreover, leveraging HRL in conjunction with causal inference models enhances RCA by uncovering latent dependencies within construction data. This approach leverages HRL to iteratively refine causal and temporal graphs, incorporating domain expertise to correct errors and optimize decision-making. By integrating human feedback into reinforcement learning frameworks, decision-makers can identify systemic root causes and proactively address issues, thereby reducing uncertainty in intricate project environments (Li et al., 2023). This approach facilitates proactive problem resolution and reduces uncertainty in complex project environments.

By incorporating human feedback into HRL-driven RCA, construction project management frameworks become more adaptive, interpretable, and aligned with real-world operational constraints. This synergy between AI-driven decision-making and expert domain knowledge fosters transparency, enhances trust among stakeholders, and improves the overall efficiency of RCA processes in construction.

Table 2.2: Summary of Relevant Literature on RLHF and Causal Analysis

Paper	Objective	Key Findings	Methodology
Bai et al. (2022)	Training language models with RLHF for helpfulness and harmlessness	RLHF improves NLP model alignment with human feedback; models trained iteratively improve efficiency	Reinforcement learning from human feedback (RLHF) with preference modeling
Shen et al. (2024)	Improving RLHF using contrastive rewards	Introduces contrastive reward penalty to enhance RLHF robustness and reduce variance	Contrastive reward model, Proximal Policy Optimization (PPO)
Lin et al. (2020)	Review of interactive RL from human social feedback	Discusses various methods for integrating human feedback into reinforcement learning to improve adaptability	Survey-based review of reinforcement learning techniques and human-agent interaction
Chen et al. (2023)	Role of feedback in AI applications, from ChatGPT to autonomous systems	Highlights the significance of human feedback in AI training across multiple domains	Analytical perspective, case study-based approach
Ouyang et al. (2022)	Training LMs to follow instructions with human feedback	InstructGPT models significantly outperform GPT-3 on instruction-following tasks	Supervised fine-tuning and RLHF
Wang et al. (2023)	Root cause analysis in microservices using hierarchical RLHF	Uses hierarchical RLHF to reduce query complexity and improve root cause discovery	Hierarchical reinforcement learning, causal graph analysis

Chapter 3

Methodology

Given a dataset of annotated text instances, where each instance contains textual event descriptions along with labeled causal and temporal relationships, our goal is to refine a given causal-temporal graph by incorporating expert knowledge. Instead of relying solely on conventional graph inference methods, we employ a Hierarchical Reinforcement Learning (HRL) framework integrated with a Human Feedback (HF) loop to update and improve causal and temporal relationships, ensuring that the graph more accurately captures real-world dependencies.

The process begins with an initial graph construction, where nodes (events) and edges (causal and temporal relationships) are extracted from the dataset using a parsing module. The graph structure is represented as an adjacency matrix with some errors. This initial graph is iteratively refined within the HRL framework. High-Level and Low-Level agents modify the graph structure and edge classifications based on expert feedback and a domain-specific reward mechanism. This combined approach enables a nuanced and accurate inference of causal and temporal relations. The overall workflow is depicted in Figure 3.1, with interconnected components of the HRL framework and the iterative graph optimization process.

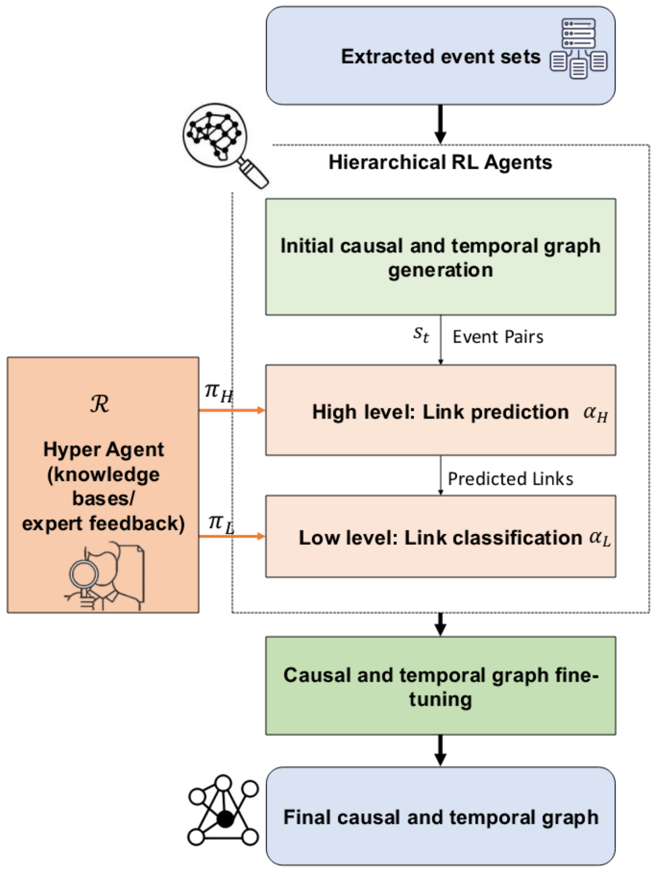

Figure 3.1: Framework of the causal and temporal discovery process. The process uses HRL with expert feedback from a hyper agent to iteratively refine a causal and temporal graph.

3.1 Hierarchical Reinforcement Learning Agents

Our Hierarchical Reinforcement Learning (HRL) approach enhances causal and temporal graph generation by employing a multi-level decision-making framework. HRL enables agents to operate at different hierarchical levels, allowing complex tasks to be decomposed into manageable sub-tasks, improving both learning efficiency and model generalization (Sutton and Barto, 2018).

In the context of Root Cause Analysis (RCA) for Requests for Information (RFIs), HRL provides a structured methodology for optimizing causal and temporal graph structures (Wang et al., 2023). The hierarchical framework consists of:

- **High-Level Agents**: Responsible for defining overarching objectives, such as identifying key causal relationships in construction projects and guiding lower-level agents towards refining RCA outputs.

- **Low-Level Agents**: Focused on executing granular tasks, such as identifying specific RFIs, mapping dependencies, and continuously refining graph representations to enhance interpretability and predictive accuracy.

This hierarchical decomposition of learning enables improved scalability and adaptability in RCA systems. Unlike traditional reinforcement learning methods, HRL introduces temporal abstraction, allowing high-level policies to dictate long-term strategies while low-level policies focus on real-time decision-making. This structure ensures that reinforcement learning agents can efficiently process complex construction data and continuously adapt to evolving project conditions (Zhang et al., 2024).

Additionally, HRL integrates expert feedback mechanisms to enhance decision-making and mitigate model biases. By incorporating domain expertise, HRL-based agents iteratively refine causal and temporal graphs, ensuring that outputs align with real-world construction constraints and improve anomaly detection. The iterative feedback loop enhances interpretability, making HRL a valuable tool for optimizing RCA workflows in large-scale construction projects (Panahi et al., 2023).

Through the integration of HRL, RCA systems benefit from increased adaptability, reduced processing overhead, and improved accuracy in predicting project risks and optimizing resolution strategies. This approach not only enhances decision-making but also contributes to minimizing construction delays and cost overruns, ultimately improving overall project performance.

3.1.1 Markov Decision Process

The HRL framework is formulated as a Markov Decision Process (MDP), which defines the environment, actions, and rewards (Fig 3.2). The MDP consists of a tuple of (S, A, P, R, γ), where:

- S is a set of states, with each state $s \in S$ representing the current configuration of causal and temporal links in the graph.
- A is a set of actions available in each state. For the high-level policy, actions $a \in \{\text{link}, \text{unlink}\}$ determine whether to add or remove a link between the events. For the low-level policy, actions $a \in \{\text{CLINK}, \text{TLINK}\}$ specify the causal and temporal relationships, respectively.
- $P(s'|s, a)$ represents the state transition probability, modeling the likelihood of reaching a new state s' given the current state s and action a.
- $R(s, a)$ is the reward function guiding the optimization process, where positive rewards reinforce correct inferences, and penalties discourage errors.
- γ is the discount factor that balances immediate versus long-term rewards, ensuring stability in learning across different hierarchical levels (Sutton and Barto, 2018).

3.2 Policies and Value Functions

HRL employs a hierarchical policy structure, where higher-level policies determine overall strategy while lower-level policies optimize specific decisions (Wang et al., 2023). This hierarchical framework allows for greater flexibility and efficiency in decision-making by enabling different levels of abstraction for problem-solving.

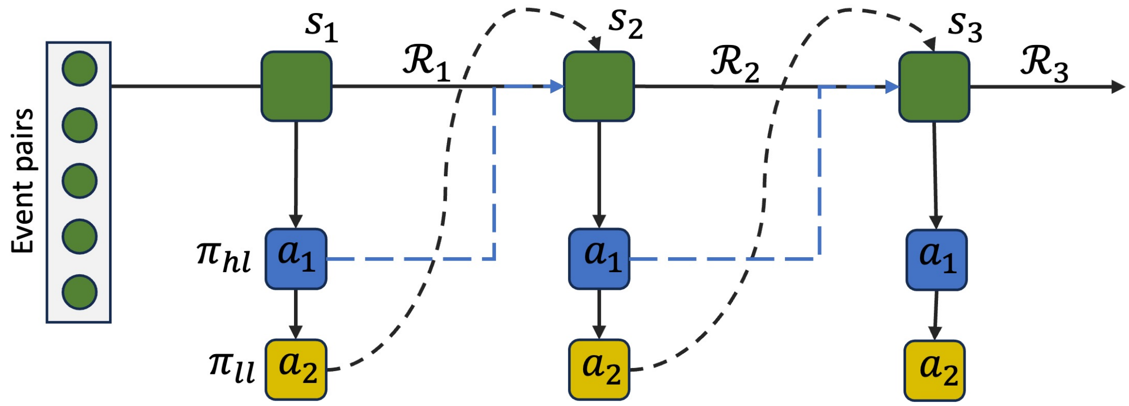

Figure 3.2: Overview of the hierarchically structured reinforcement learning, which consists of high-level and low-level policy.

The primary function of the value function $V(s)$ is to estimate the expected cumulative reward from a given state. This function is crucial for guiding HRL agents in making optimal decisions, ensuring that they move towards more effective representations of causal and temporal relationships. It is defined as:

$$V^{\pi}(s) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_0 = s, \pi\right] \tag{1}$$

where $R(s_t, a_t)$ is the immediate reward received after taking action a_t in state s_t, and $\gamma \in [0, 1]$ is the discount factor that determines the weight of future rewards. The expectation is taken over trajectories generated by following policy π.

The state-action function $Q(s, a)$ plays a complementary role by evaluating the impact of taking action a in state s (Sutton and Barto, 2018). This enables the agent to determine the most beneficial actions within its environment based on learned experience and feedback mechanisms.

$$Q^{\pi}(s, a) = \mathbb{E}\left[\sum_{t=0}^{\infty} \gamma^t R(s_t, a_t) \mid s_0 = s, a_0 = a, \pi\right] \tag{2}$$

This function evaluates the expected future rewards when taking action a in state s and subsequently following policy π.

Furthermore, value functions in HRL are updated iteratively using dynamic programming or temporal difference learning. These methods allow for continuous improvement of policy effectiveness by adjusting estimated rewards and state values over time. Through this structured learning approach, HRL agents become increasingly adept at recognizing patterns, predicting anomalies, and refining causal and temporal graphs to enhance decision-making accuracy.

3.2.1 Policy Optimization in HRL

In HRL, policies are optimized through iterative updates based on experience and reinforcement learning signals. The agent seeks to find an optimal policy π^* that maximizes the expected reward:

$$\pi^* = \arg\max_{\pi} V^{\pi}(s) \tag{3}$$

Policies can be either deterministic ($\pi(s) = a$) or stochastic ($\pi(a|s)$), where stochastic policies allow for exploration, helping the agent discover better strategies in complex decision-making environments.

3.2.2 Value Function Updates

HRL agents refine their value functions using dynamic programming techniques or temporal difference (TD) learning. The Bellman equation provides a recursive relationship for value function updates:

$$V(s) = \max_{a} \left[R(s, a) + \gamma \sum_{s'} P(s'|s, a)V(s') \right] \tag{4}$$

Similarly, Q-values are updated using Q-learning.

Q-learning is a fundamental reinforcement learning algorithm used to estimate the optimal action-value function for a given environment. It operates by iteratively updating the Q-values based on observed rewards and the estimated future rewards. The goal of Q-learning is to learn a policy that maximizes the expected cumulative reward by updating the action-value function using the Bellman equation. The update rule is given as:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left(r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right) \tag{5}$$

where

- $Q(s,a)$ is the current estimate of the action-value function, representing the expected reward for taking action in state,
- $\max_{a'} Q(s',a')$ represents the highest Q-value for the next state , which helps the agent choose the best future action.,
- α is the learning rate, controls how much new information overrides the old information in the Q-value update,
- r is the immediate reward received after executing an action in state, and
- γ the discount factor, determines the importance of future rewards compared to immediate rewards.

Learning Rate (r): The learning rate determines how much new information overrides old information when updating Q-values. A high learning rate (close to 1) allows for rapid learning but may cause instability, while a low learning rate (close to 0) results in slow convergence but ensures more stable learning.

Discount Factor (γ): The discount factor represents the weight given to future rewards. A discount factor close to 1 prioritizes long-term rewards, making the agent more future-oriented. Conversely, a lower discount factor makes the agent focus more on immediate

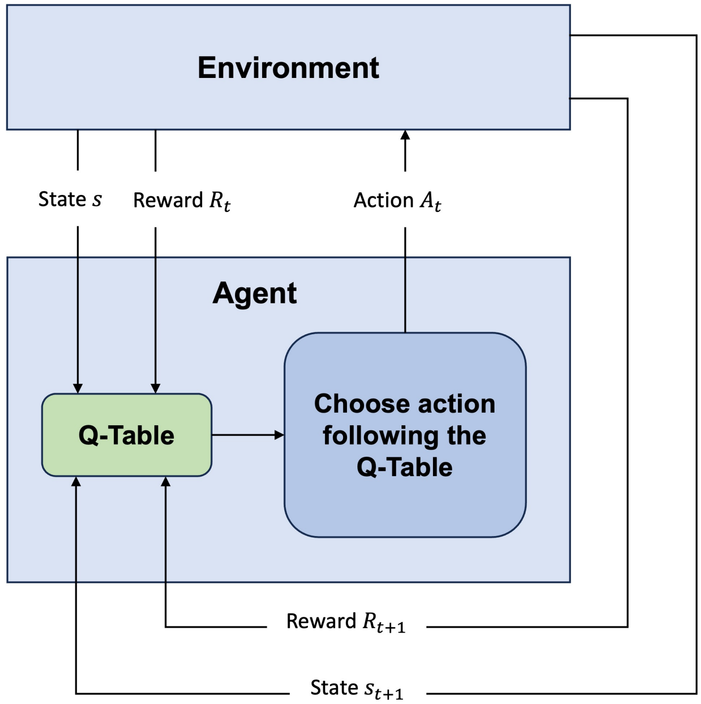

Figure 3.3: Q-learning overview

rewards, which is useful in environments with short-term goals. Choosing appropriate values for r and γ is crucial for ensuring efficient and effective learning in reinforcement learning problems.

Q-learning is widely used in reinforcement learning because it enables agents to learn optimal decision-making policies without requiring prior knowledge of the environment's transition probabilities. It follows an off-policy learning approach, meaning the agent can learn from previously collected experience without strictly following the current policy. This characteristic makes Q-learning particularly useful in scenarios where the environment is complex, stochastic, or partially observable.

The key benefits of Q-learning include:

- **Model-free learning**: Q-learning does not require knowledge of the environment's dynamics (i.e., transition probabilities), making it applicable to a wide range of problems.

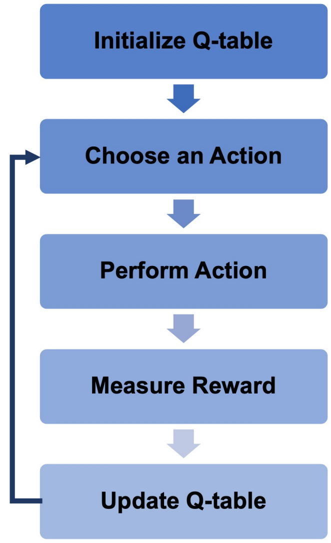

Figure 3.4: Q-learning process

- **Convergence to the optimal policy**: Given sufficient exploration and an appropriate learning rate, Q-learning is guaranteed to converge to the optimal Q-values, allowing the agent to make the best decisions.

- **Versatility**: Q-learning can be applied to a variety of reinforcement learning problems, from game-playing to robotic control and autonomous decision-making.

While more advanced RL algorithms such as Deep Q-Networks (DQN), Fitted Q-Iteration (FQI), Deep Deterministic Policy Gradient (DDPG), or Direct Policy Optimization (DPO) offer improvements in complex function approximation and continuous action spaces, they were not employed in this study due to several reasons. First, traditional Q-learning is sufficient for the discrete decision-making process in causal and temporal graph optimization, where the action space is relatively small (linking or unlinking events, classifying links). Second, the interpretability of tabular Q-learning provides clear decision-making rationales, which is crucial for ensuring transparency and trust in construction project analysis. Lastly, deep RL methods require extensive training data and computational resources, which may not be justified given the current dataset size and task complexity. Future work may explore hybrid approaches incorporating function approximation techniques for scalability while retaining the advantages of structured decision-making in hierarchical RL.

3.2.3 High-Level Agents

High-level agents are responsible for predicting and managing the structural relationships between events in a graph by determining whether connections should be established or removed. This decision-making process follows a Markov Decision Process (MDP), where the state space represents the current configuration of causal and temporal links, and the action space consists of operations such as linking or unlinking nodes. By iteratively refining the graph structure, these agents enhance the model's understanding of event dependencies and improve knowledge representation.

The high-level agent operates within the MDP framework, defined as a tuple (S, A, P, R, γ), where:

- **State Space** (S): Each state $s \in S$ represents a graph configuration at a given timestep, characterized by:

  $$s = G_t = (V, E_t) \tag{6}$$

  where V is the set of event nodes, and E_t is the set of existing edges (causal or temporal links) at time t.

- **Action Space** (A): The agent selects an action $a \in A$ that modifies the graph by either:

  $$A = \{\text{link}(u, v), \text{unlink}(u, v)\} \tag{7}$$

  where $u, v \in V$.

- **Transition Probability** ($P(s'|s,a)$): The environment responds to an action by transitioning to a new state $s' = G_{t+1}$:

$$P(s'|s,a) = P(G_{t+1}|G_t, a) \tag{8}$$

- **Reward Function** ($R(s,a)$): The agent receives a reward signal, defined as:

$$R(s,a) = w_1 R_{\text{struct}} + w_2 R_{\text{accuracy}} + w_3 R_{\text{consistency}} \tag{9}$$

- **Discount Factor** (γ): The discount factor balances short-term and long-term rewards.

To achieve optimal link prediction, high-level agents employ reinforcement learning techniques such as Q-learning, policy gradient methods, or actor-critic approaches. These agents are trained using an exploration-exploitation framework, where they iteratively assess the potential impact of modifying links based on observed rewards. Rewards may be determined by factors such as structural consistency, predictive accuracy, and alignment with ground truth data. By continuously learning from interaction feedback, the high-level agents dynamically adapt to evolving data patterns, ensuring the integrity of causal and temporal relationships.

Beyond simple link addition or removal, high-level agents also assess the long-term impact of structural adjustments, ensuring that changes contribute to an improved understanding of event causality. Through strategic decision-making, these agents refine the graph structure in a manner that balances exploration (discovering new meaningful connections) and exploitation (preserving reliable, validated links). Their operation facilitates efficient event dependency modeling, which is critical for complex analytical applications such as root cause analysis, anomaly detection, and predictive event forecasting.

3.2.4 Low-Level Agents

Low-level agents operate in conjunction with high-level agents, functioning as specialized classifiers that refine and categorize the established connections. Their primary role is to assign appropriate link types to newly formed relationships, distinguishing between causal links (CLINK) and temporal links (TLINK). These agents also fine-tune connection details to enhance accuracy, ensuring that the assigned relationships adhere to domain-specific constraints and are semantically meaningful.

Unlike high-level agents, which focus on structural modifications, low-level agents engage in classification tasks that leverage machine learning models, supervised learning techniques, and reinforcement learning strategies. They extract features from event descriptions, evaluate contextual relationships, and assess factors such as time dependencies, co-occurrence patterns, and causal indicators. By incorporating domain knowledge and training data, low-level agents optimize the assignment of link types, improving the reliability of the generated event graph.

Low-level agents operate in conjunction with high-level agents, functioning as specialized classifiers that refine and categorize the established connections. Their role is to assign the appropriate link types (Causal or Temporal) once a connection is made. The key components of low-level agent decision-making include:

- **State Space (S')**: The state s' represents the graph after the high-level agent has decided to modify an edge.
- **Action Space (A')**: The low-level agent classifies links by choosing:

$$A' = \{\text{CLINK}, \text{TLINK}\} \tag{10}$$

- **Reward Function ($R(s', a')$)**: The agent receives feedback based on classification accuracy, with penalties for misclassification.

A continuous feedback loop between high-level and low-level agents ensures that the graph remains both structurally valid and semantically precise.

To further refine link classification, low-level agents implement confidence scoring mechanisms and parameter adjustments. These refinements involve adjusting time intervals, weighting causal influence, and enhancing the interpretability of inter-event relationships. The continuous feedback loop between high-level and low-level agents ensures that the graph remains structurally sound while accurately capturing the complexity of event interactions.

By operating within a hierarchical framework, the interplay between high-level and low-level agents enhances the robustness of the learned representations. This multi-tiered approach facilitates more precise event linking, leading to improved decision-making in domains such as automated RFI analysis, intelligent document processing, and predictive modeling of real-world systems.

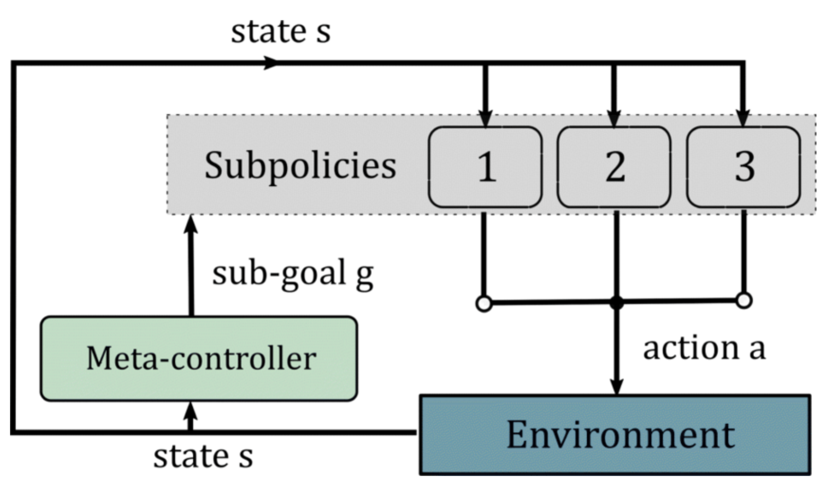

Figure 3.5: The hierarchical reinforcement learning framework.

3.3 Reward Model for Graph Optimization

The reward model plays a critical role in reinforcement learning by evaluating the accuracy and efficiency of the predicted graph structure compared to a ground-truth reference.

This dataset (reference) consists of expert-annotated causal and temporal relationships that serve as a reference for evaluating the accuracy of the generated graph. In real-world RFI workflows, an initial causal graph would typically be constructed by junior engineers based on project documentation and correspondence, with senior experts reviewing and refining the relationships. However, in our experimental setup, we simulate this process synthetically by generating randomly downgraded graphs from expert-annotated datasets. These degraded graphs introduce errors, mimicking the incomplete or incorrect annotations typically made by less experienced engineers. The reinforcement learning model then iteratively improves these graphs, guided by a reward function that reinforces correct inferences and penalizes incorrect relationships. This approach ensures that the learned causal and temporal relationships align with expert knowledge, improving the accuracy and interpretability of the generated event graphs.

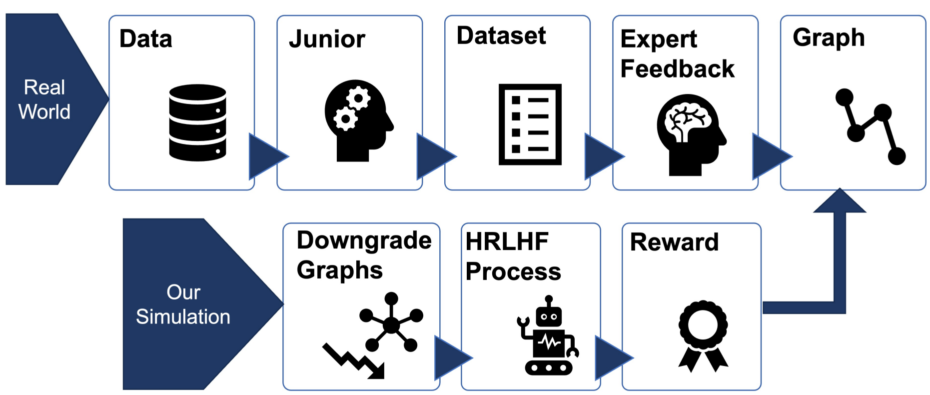

Figure 3.6: Our experimental setup and real-world.

It provides feedback to the agents, enabling them to refine their decisions over time and improve link prediction. The reward model considers multiple factors in its evaluation, ensuring that the learned graph structure aligns with domain requirements and remains structurally coherent.

To achieve this, the reward function is designed to balance correct link prediction and error minimization through a set of defined metrics:

- **Correct Link Rewards or True Positive Rewards (TP):** The model assigns positive rewards for correctly predicted causal (CLINK) and temporal (TLINK) links that match the ground-truth graph, which indicate that the model has successfully identified valid relationships between events.

- **False Positive Penalties (FP):** Incorrectly added links that do not exist in the true graph receive negative penalties to discourage over-linking.

- **False Negative Penalties (FN):** Missing links that should exist in the graph are penalized to ensure that all valid relationships are captured.

  both of which can significantly impact downstream applications relying on the graph.

- **Graph Density Regulation ($P_{\mathbf{D}}$):** To prevent excessive linking, that could lead to unnecessary complexity, the reward model introduces a density penalty when the graph exceeds a predefined threshold, ensuring that the structure remains interpretable.

- **Maintenance Rewards ($R_{\mathbf{M}}$):** A small reward is given for maintaining existing correct links, promoting stability and preventing unnecessary modifications. This ensures that once a valid causal or temporal link is established, it remains in the refined graph unless strong evidence suggests otherwise.

The formulation of the reward function ensures that the reinforcement learning agents can iteratively refine the graph in a structured and optimal manner. By integrating expert knowledge, penalizing inaccuracies, and maintaining structural efficiency, the model effectively drives the learning process toward producing high-quality causal-temporal graphs that accurately reflect real-world dependencies.

Mathematically, the total reward for a given graph is computed as:

$$R = \sum_i TP_{\text{CL}} i \cdot w_{\text{CL}} + \sum_j TP_{\text{TL}} j \cdot w_{\text{TL}} - \sum_k \text{FP} k \cdot w_{\text{fp}} - \sum_m \text{FN} m \cdot w_{\text{fn}} + R_{\text{M}} - P_{\text{D}} \quad (11)$$

where each component is defined as:

$$R_{\text{M}} = (N_{\text{CL}} + N_{\text{TL}}) \times w_{\text{M}}, \quad (12)$$

$$P_{\text{D}} = \begin{cases} 0 & \text{if } \rho \leq \rho_{\text{t}}, \\ w_{\text{D}} \times (1 + 5 \times (\rho - \rho_{\text{t}})) & \text{if } \rho > \rho_{\text{t}}, \end{cases} \quad (13)$$

where w represents the weight parameter for each reward and penalty component, w_{D} is the specific penalty applied to discourage excessive graph density, ρ denotes the current graph density, and ρ_{t} is the threshold beyond which density penalties are introduced.

The inclusion of a density penalty is crucial for maintaining an optimal graph structure. Without this regulation, the model may tend to over-link nodes in an attempt to maximize reward signals, leading to an unnecessarily complex and cluttered graph. Over-linking increases noise, making it difficult to discern meaningful causal and temporal relationships. The density penalty P_{D} is formulated to dynamically scale based on how much the graph exceeds the predefined threshold ρ_{t}, ensuring that only essential edges are retained while redundant or spurious connections are penalized. The coefficient of 5 in the penalty function was determined empirically to provide a balance between encouraging necessary link formation and penalizing excessive edge additions. This coefficient ensures a gradual yet firm discouragement of over-linking as the graph density surpasses the threshold.

To determine the optimal reward parameters, extensive empirical tuning was conducted. Various weight settings were tested across multiple experimental iterations to identify the best trade-off between reinforcing correct classifications, discouraging incorrect inferences, and regulating graph complexity. A key challenge in designing the weight parameters was ensuring that penalties did not overly restrict edge formations while still preventing model overfitting to noisy data. The final set of weight parameters reflects a balance between model interpretability and accuracy. Table 3.1 provides a summary of the reward and penalty weights used in the experiments, ensuring reproducibility and guiding future refinements of the model.

Table 3.1: Hyperparameters for Reward Function

Parameter	Value	Description
$w_{\text{CL/TL}}$	100	Reward for correctly classified CLINKs
w_{M}	50	Reward for maintaining correct edges
$w_{\text{FP/FN}}$	-100	Penalty for false positives (spurious edges)
w_{D}	-20	Penalty for excessive graph density
ρ_{t}	0.4	Graph density threshold

These values were determined through 30 carefully designed experiments, where each iteration was used to analyze the model's performance under different reward and penalty configurations. The experiments involved evaluating the model's ability to generalize across multiple datasets, ensuring that the reinforcement learning agents refined the graph optimally without introducing redundant connections. By systematically adjusting weight parameters and monitoring performance metrics such as precision, recall, and F1-score, the reward model was fine-tuned to maintain a balance between learning efficiency and graph accuracy. The iterative refinement process helped prevent both underfitting—where essential causal and temporal relationships were missed—and overfitting, where excessive link formation led to inflated graph complexity. Ultimately, this empirical tuning process ensured that the model consistently converged to an optimal structure, reinforcing the accurate identification of causal dependencies while minimizing classification errors.

3.4 Training and Testing Workflow

The training and testing workflow ensures that the hierarchical reinforcement learning model efficiently learns to optimize link prediction in event graphs. The process as have shown in Fig 3.7 is designed to simulate real-world conditions, allowing the model to adapt to various complexities and uncertainties in event relationships. The training phase involves iterative learning and refinement, while the testing phase evaluates the model's generalization ability using unseen data.

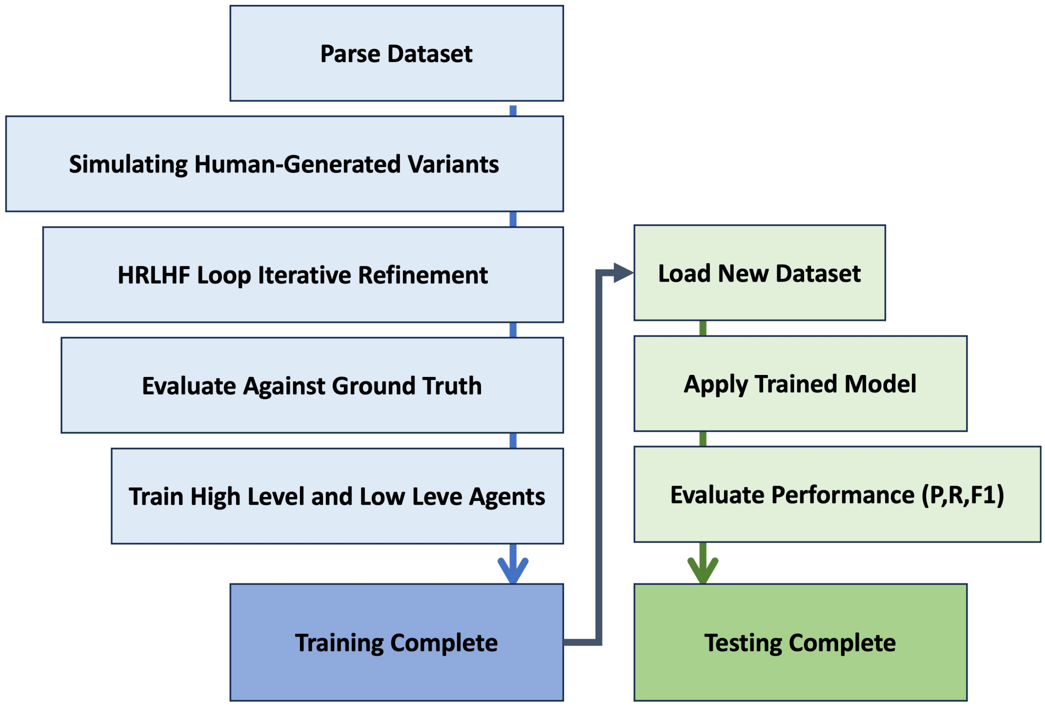

Figure 3.7: Train and test workflow.

3.4.1 Training Phase

The training phase consists of four key steps that allow the model to learn how to structure the event graph accurately.

Step 1: Converting Event Data into a Directed Graph

The first step involves parsing event data and representing it as a directed graph. This step extracts events and their relationships, forming the initial structure that will be refined through reinforcement learning. The graph includes nodes representing events and edges corresponding to causal (CLINK) or temporal (TLINK) relationships.

Step 2: Simulating human-generated variants

To simulate real-world uncertainties and test the model's robustness, intentional errors are introduced into the graph. This step distorts the ground-truth relationships by randomly removing, adding, or modifying links. The purpose of this step is to expose the model to noisy data, forcing it to learn corrective strategies that improve the accuracy of link predictions.

Step 3: Iterative Refinement Using HRLHF

The core of the training process is the HRLHF loop, where high-level and low-level agents iteratively refine the graph. The high-level agent decides whether to link or unlink nodes, while the low-level agent classifies the type of relationship. Reinforcement learning drives these decisions, with agents receiving rewards based on the accuracy of their modifications. Over multiple iterations, the agents learn to optimize link structures by maximizing correct predictions while minimizing false links.

Step 4: Performance Evaluation

After the refinement process, the model's performance is evaluated against ground-truth data. This function measures how well the refined graph matches the true event structure using precision, recall, and F1-score. A well-trained model should demonstrate a high degree of accuracy in reconstructing the original event graph, indicating successful learning.

3.4.2 Testing Phase

Once training is complete, the trained model is applied to new, unseen datasets to evaluate its generalization performance.

Step 1: Applying the Model to a New Dataset

In the testing phase, the trained reinforcement learning model is applied to previously unseen event graphs. This step assesses whether the model can accurately predict causal and temporal relationships in data that was not included in training. The model operates under the same decision-making framework as in training, using learned policies to refine graph structures.

Step 2: Measuring Model Improvement

To quantify the effectiveness of the model, its performance is measured using standard evaluation metrics: precision, recall, and F1-score. Precision assesses how many of the predicted links are correct, recall measures the proportion of actual links that were correctly identified, and F1-score provides a balanced measure of both. A significant improvement in these metrics compared to the initial noisy graph demonstrates the effectiveness of the reinforcement learning process.

By following this structured training and testing workflow, the hierarchical reinforcement learning framework ensures continuous improvement in link prediction accuracy. The iterative refinement process enables the model to adapt to real-world complexities, making it a robust approach for analyzing and structuring event-driven datasets.

3.5 Human Feedback Integration

This framework incorporates HF through a ground truth reference graph, which encodes expert-validated causal and temporal relationships between events. This reference graph serves as the benchmark for evaluating generated graphs during training. A reward model (Equation 2) quantifies the similarity between the evolving graph and the ground truth, guiding HRL agents in refining the structure iteratively. Starting with an initial graph with some errors, RL agents utilize this feedback-driven reward system to improve accuracy.

During the training phase, the ground truth graph is directly integrated into the reward model, providing feedback at each iteration. In this feedback loop, the RL agent modifies the initial graph by adding, removing, or reclassifying edges to maximize the cumulative reward. The ground truth graph is no longer used for direct feedback during testing. Instead, the trained RL model, having already internalized structural patterns from the ground truth, operates to infer causal and temporal links independently. The system applies the learned policy to new, unseen datasets without explicit ground truth comparisons. As illustrated in Figure 3.1, the ground truth graph influences training by guiding high-level and low-level agents. The reward model, informed by the ground truth, ensures that the graph evolves toward a more accurate representation of causal dependencies. Refining the reward function and explicitly integrating expert-validated structures ensures that HF is leveraged to enhance causal and temporal inference.

Chapter 4

Experiments and Results

4.1 Experimental Setup

4.1.1 Datasets Details

We evaluate our HRLHF model on the widely adopted Causal-TimeBank dataset (Mirza et al., 2014), a widely recognized benchmark dataset for causal and temporal event relation extraction. This dataset extends the TimeBank corpus (Pustejovsky et al., 2003) with additional causal annotations, making it one of the most comprehensive resources for studying event interactions. It has been widely adopted in natural language processing (NLP) and information extraction research, serving as a standard benchmark for various models in causal and temporal reasoning. The Causal-TimeBank dataset consists of:

- 184 documents, extracted from diverse sources such as news articles, historical texts, legal documents, and medical reports.
- 6,813 annotated events, representing key occurrences described in the text.
- 318 event pairs annotated with causal relations (CLINKs), explicitly defining cause-effect relationships between events.

- 5,118 event pairs annotated with temporal relations (TLINKs), describing the chronological ordering of events.

The dataset covers multiple domains, ensuring broad applicability across various fields. The combination of causal and temporal annotations makes it particularly valuable for applications in event prediction, automated reasoning, and knowledge graph construction. While this study focuses on event graph optimization from structured RFI text, the proposed method aligns with digital construction workflows. In practice, RFIs are commonly managed through structured project platforms such as Aconex or Procore, where textual communication is stored in a digital format. Our approach assumes that event data are available in digital form. The initial event graphs can then be constructed manually or semi-automatically and passed to the HRLHF model for optimization.Following the standard procedure in causal and temporal relation extraction tasks, we treat the annotated causal and temporal links as the target relationships. The dataset follows a rigorous manual annotation process, where expert linguists labeled event relations based on linguistic cues and domain-specific knowledge. The dataset includes:

Causal Relations (CLINKs) define direct cause-effect relationships between events.
Example: *"The earthquake caused widespread destruction."* Cause: "earthquake" Effect: "destruction". CLINKs are annotated based on explicit causal markers (e.g., "because", "due to") and inferred causal dependencies.

Temporal Relations (TLINKs) define the chronological order of events (e.g., Before, After, Simultaneous).
Example: *"The storm hit the city before power was restored."* Event 1: "storm hit the city" Event 2: "power was restored" TLINK: Before. Annotations follow the TimeML standard, which categorizes event ordering into Before, After, Simultaneous, and Vague labels.

The Causal-TimeBank dataset is extensively used in the research community to evaluate models for causal and temporal event reasoning. The dataset's structured nature makes it a critical resource for training and evaluating models that aim to automate causal and temporal reasoning in real-world applications such as disaster response, medical diagnosis, legal text analysis, and historical event prediction.

4.1.2 RFI Data Alignment with the Causal-TimeBank Annotation Format

The structured nature of RFIs aligns closely with event-based annotated datasets like the Causal-TimeBank. Each RFI represents an issue, a need for clarification, and a resolution, often following a well-defined temporal and causal sequence. This characteristic makes RFIs suitable for annotation using methods developed in linguistic annotation tasks, enabling automated extraction and analysis (Mirza and Tonelli, 2014).

To facilitate the annotation of RFIs, the CELCT Annotation Tool (CAT) can be employed. CAT is a web-based tool designed for linguistic and semantic annotation, supporting multi-layer annotations and structured data extraction. Originally developed for annotating temporal events following the It-TimeML specifications (Caselli et al., 2011), CAT is flexible enough to be adapted for RFI annotation by capturing event sequences, causal dependencies, and temporal links.

Building on the structured annotation approach of the Causal-TimeBank dataset (Mirza and Tonelli, 2014), RFIs can be effectively annotated due to their inherent similarities with event-based datasets. Both RFIs and Causal-TimeBank data capture event sequences with explicit causal and temporal dependencies, making the mapping between them intuitive and systematic. The following components highlight how RFIs align with the structure of Causal-TimeBank:

- **Event Identification**: Similar to annotated events in Causal-TimeBank, RFIs correspond to distinct project events, including submission, response, and resolution. These milestones can be tagged as separate events, providing a structured representation of project inquiries.

- **Causal Relations**: RFIs document causal relationships between issues and their resolutions, similar to the causal links annotated in TimeML. These dependencies can be explicitly labeled using predefined causal relation types, mirroring the structured annotation in Causal-TimeBank (Pustejovsky et al., 2005).

- **Temporal Dependencies**: Just as TimeML captures event timelines, RFIs inherently follow a chronological sequence from submission to resolution. Timestamp annotations allow for precise representation of these dependencies, preserving the temporal flow of information (Verhagen et al., 2006).

- **Action and Response Mapping**: RFIs establish structured interactions between stakeholders, similar to the discourse-level annotations in TimeML. By annotating the connections between RFI queries and responses, we can uncover communication patterns and decision-making structures within construction projects (Lenzi et al., 2012).

To facilitate this annotation process, CAT (Content Annotation Tool) provides a structured framework for systematically labeling RFIs. By leveraging its built-in inter-annotator agreement measurement and XML-based structured output, CAT ensures consistency, scalability, and efficiency in managing large-scale annotated RFI datasets. This alignment with established event-based annotation methodologies allows RFIs to be effectively processed for AI-driven classification and automated analysis (Lenzi et al., 2012).

This structured annotation approach enhances automated analysis by enabling AI-driven classification, response prioritization, and predictive modeling. As demonstrated in event-based annotation frameworks such as TimeML, the integration of causal and temporal dependencies in RFIs leads to more interpretable and reliable information extraction models (Mirza and Tonelli, 2014).

4.1.3 Implementing RFI Annotation Using the Causal-TimeBank Dataset Format

To annotate RFIs effectively, the first step is to digitize and preprocess the documents. This involves converting them into a machine-readable format and standardizing the text to ensure consistency. Cleaned and structured data is essential for accurate annotation and automated processing.

Key events within the RFI lifecycle must then be identified. Each RFI contains significant actions such as submission, response, and resulting decisions. These events are tagged using the label, with attributes assigned for event type (e.g., request, response), tense (past, present, future), and project phase (design, construction).

Time expressions must also be annotated to capture the sequence of events. Using the tag, references to specific dates (e.g., “March 15, 2025”), durations (e.g., “within two weeks”), and general timeframes (e.g., “after approval”) are recorded to establish temporal relationships between events.

To define dependencies, causal and temporal links are added:

Temporal Links ($< TLINK >$): Describe the sequence of events (before, after, during). For example, linking an RFI submission to its response deadline clarifies response expectations.

Causal Links ($< CLINK >$**)**: Identify cause-and-effect relationships, such as an RFI request leading to a project design change.

Signal Words ($< C-signal >$**)**: Terms like “because,” “therefore,” or “after” help in marking explicit causal and temporal dependencies.

Finally, annotation tools like BRAT or GATE are used to manually tag RFI texts following the TimeML format. These tools streamline the annotation process, ensuring consistency and accuracy. By applying this structured approach, RFIs can be effectively analyzed, improving classification, response prioritization, and predictive modeling for better project management.

4.1.4 Causal-TimeBank and RFI Comparison

They share several foundational attributes that make them highly compatible for use. Techniques used to parse and analyze the Causal-TimeBank could be adapted for RFI analysis (Natural language processing techniques such as entity recognition, dependency parsing, and relationship extraction). RFIs typically include cause-and-effect questions within construction projects, implying a need for recognizing and processing causal relationships within the inquiries and responses. Causal-TimeBank explicitly annotated for causal relationships between events. The focus on both temporal and causal annotations in Causal-TimeBank is particularly relevant to RFI datasets where the timing and sequence of construction events, decisions, and queries play a crucial role in project management.

The Causal TimeBank dataset includes two primary types of relationships to denote directional interactions:

Table 4.1: Comparison between RFI and Causal-TimeBank dataset

	RFI	**Causal-TimeBank**
Content	Text	Text
Relationships	Cause-and-effect	Causal links between events
Domain	Specific aspects	News stories
Causality	Contains causal queries	Explicit causal annotation
Structure	Formal language	Formal language
Complexity	Complex sentences	Complex sentences
Temporal Info	Project schedules	Event sequencing
Links	A leads to B	A leads to B

E-E pair	**Sentence**	**TE3-gold**	**TE-label**	**CA-label**	**Post-editing**
(e_{32}, e_{44})	*The [incident]* $_{e_{32}}$ *provoked an international [outcry]* $_{e_{44}}$*...*	-	SIMULTANEOUS	CLINK	BEFORE
(e_{32}, e_{45})	*The [incident]* $_{e_{32}}$ *provoked an international outcry and led to a major [deterioration]* $_{e_{45}}$ *in relations...*	-	AFTER	CLINK	BEFORE
(e_{18}, e_{19})	*...the [inspections]* $_{e_{18}}$ *were directly linked to the new law on NGOs and the targeted groups' [compliance]* $_{e_{19}}$ *with it.*	-	IS_INCLUDED	CLINK-R	AFTER
(e_{4}, e_{6})	*A haze akin to volcanic fumes [cloaked]* $_{e_{4}}$ *the capital, causing convulsive [coughing]* $_{e_{6}}$ *and...*	INCLUDES	AFTER	CLINK	BEFORE

Figure 4.1: Causal-TimeBank dataset before annotation.

TLINK (Temporal Link): This link type encapsulates temporal relationships with various possible statuses such as *Before*, *After*, *Includes*, *Is_Included*, among others. An example of a TLINK entry is shown below:

$< TLINK\ comment = “”\ id = “1”\ relType = “BEFORE” >$

$< source\ id = “1” / >$

$< target\ id = “0” / >$

$< /TLINK >$

In this example, `target id="0"` refers to the date on which the news was published.

CLINK (Causal Link): This represents causal relationships where the direction is from a source (cause) to a target (effect). Below is an example of a CLINK:

$< CLINK\ c - signalID = “10”\ comment = “”\ id = “12” >$

$< source\ id = “11” / >$

$< target\ id = “8” / >$

$< /CLINK >$

```
<Document doc_name="ABC19980108.1830.0711">
  <Markables>
    <TIMEX3 RELATED_TO="" TAG_DESCRIPTOR="Empty_Mark" beginPoint="" comment="" endPoint=""
freq="" functionInDocument="CREATION_TIME" id="0" quant="" type="DATE" value="1998-01-08" />
    <EVENT aspect="PROGRESSIVE" certainty="" class="OCCURRENCE" comment="" factuality="" id="1"
modality="NONE" polarity="POS" pos="VERB" tense="PRESENT">
      <token_anchor id="8" />
    </EVENT>
    <TIMEX3 beginPoint="" comment="" endPoint="" freq="" functionInDocument="NONE" id="2"
quant="" type="DURATION" value="P1W">
      <token_anchor id="16" />
    </TIMEX3>
    <EVENT aspect="NONE" certainty="" class="OCCURRENCE" comment="" factuality="" id="3"
modality="NONE" polarity="POS" pos="NOUN" tense="NONE">
      <token_anchor id="22" />
    </EVENT>
    <EVENT aspect="PROGRESSIVE" certainty="" class="OCCURRENCE" comment="" factuality="" id="4"
modality="NONE" polarity="NEG" pos="VERB" tense="PRESENT">
      <token_anchor id="34" />
    </EVENT>
    <TIMEX3 anchorTimeID="0" beginPoint="" comment="" endPoint="" freq=""
functionInDocument="NONE" id="5" quant="" type="DURATION" value="P1D">
      <token_anchor id="37" />
      <token_anchor id="38" />
      <token_anchor id="39" />
      <token_anchor id="40" />
      <token_anchor id="41" />
    </TIMEX3>
    <EVENT aspect="PERFECTIVE" certainty="" class="OCCURRENCE" comment="" factuality="" id="6"
modality="NONE" polarity="POS" pos="VERB" tense="PRESENT">
      <token_anchor id="51" />
    </EVENT>
    <EVENT aspect="PERFECTIVE" certainty="" class="OCCURRENCE" comment="" factuality="" id="7"
modality="NONE" polarity="POS" pos="VERB" tense="PRESENT">
      <token_anchor id="60" />
    </EVENT>
    <EVENT aspect="NONE" certainty="" class="OCCURRENCE" comment="" factuality="" id="8"
modality="NONE" polarity="POS" pos="VERB" tense="PAST">
      <token_anchor id="72" />
    </EVENT>
    <TIMEX3 beginPoint="" comment="" endPoint="" freq="" functionInDocument="NONE" id="9"
quant="" type="DURATION" value="P5Y">
      <token_anchor id="74" />
      <token_anchor id="75" />
    </TIMEX3>
    <TIMEX3 beginPoint="" comment="" endPoint="" freq="" functionInDocument="NONE" id="10"
quant="" type="DURATION" value="P4Y">
```

Figure 4.2: Annotated Causal-TimeBank dataset.

Here, `source id="11"` is identified as the cause, whereas `target id="8"` is designated as the effect.

4.1.5 Using Causal-TimeBank for RFI Analysis

For texts describing events, knowing which events are important and linking them in a temporal-causal structure would allow the automatic generation of a timeline-style summary (Mirza, 2016). The Causal-TimeBank dataset and the RFI dataset, despite their differing contexts—news stories and construction projects, respectively—share several foundational attributes that make them highly compatible for use in models concerned with causal relationship analysis. Here's a detailed comparison to illustrate their relevance:

```
<Document doc_name="RFI_423">
  <Markables>
    <TIMEX3 RELATED_TO="" TAG_DESCRIPTOR="Empty_Mark" beginPoint="" comment=""
endPoint="" freq="" functionInDocument="CREATION_TIME" id="0" quant="" type="DATE"
value="2023-07-15">
      <token_anchor id="10">July 15, 2023</token_anchor>
    </TIMEX3>
    <TIMEX3 beginPoint="" comment="" endPoint="" freq="" functionInDocument="NONE"
id="1" quant="" type="DATE" value="2023-07-22">
      <token_anchor id="20">July 22, 2023</token_anchor>
    </TIMEX3>
    <EVENT aspect="NONE" certainty="" class="TASK" comment="" factuality="" id="2"
modality="NONE" polarity="POS" pos="VERB" tense="PRESENT">
      <token_anchor id="70">installed</token_anchor>
    </EVENT>
    <EVENT aspect="NONE" certainty="" class="TASK" comment="" factuality="" id="3"
modality="NONE" polarity="POS" pos="VERB" tense="FUTURE">
      <token_anchor id="86">install</token_anchor>
    </EVENT>
    <EVENT aspect="NONE" certainty="" class="TASK" comment="" factuality="" id="4"
modality="NONE" polarity="POS" pos="VERB" tense="PRESENT">
      <token_anchor id="97">avoid</token_anchor>
    </EVENT>
    <EVENT aspect="NONE" certainty="" class="TASK" comment="" factuality="" id="5"
modality="NONE" polarity="POS" pos="VERB" tense="FUTURE">
      <token_anchor id="102">follow</token_anchor>
    </EVENT>
    <EVENT aspect="NONE" certainty="" class="TASK" comment="" factuality="" id="6"
modality="NONE" polarity="POS" pos="VERB" tense="FUTURE">
      <token_anchor id="110">install</token_anchor>
    </EVENT>
    <EVENT aspect="NONE" certainty="" class="TASK" comment="" factuality="" id="7"
modality="NONE" polarity="POS" pos="VERB" tense="PRESENT">
      <token_anchor id="80">rerouted</token_anchor>
    </EVENT>
    <EVENT aspect="NONE" certainty="" class="TASK" comment="" factuality="" id="8"
modality="NONE" polarity="POS" pos="VERB" tense="PRESENT">
      <token_anchor id="90">coordinated</token_anchor>
    </EVENT>
    <EVENT aspect="NONE" certainty="" class="TASK" comment="" factuality="" id="9"
modality="NONE" polarity="POS" pos="VERB" tense="PRESENT">
      <token_anchor id="100">documented</token_anchor>
    </EVENT>
    <EVENT aspect="NONE" certainty="" class="TASK" comment="" factuality="" id="10"
modality="NONE" polarity="POS" pos="VERB" tense="FUTURE">
      <token_anchor id="198">install</token_anchor>
```


Figure 4.3: Annotated RFI dataset.

Textual Nature and Formal Structure: Both datasets are primarily composed of text that adheres to a formal structure, which is beneficial for computational processing and analysis. This formalism ensures that the data can be systematically parsed and interpreted by models designed to extract or understand causal relationships.

Causal Relationships: A central theme in both datasets is the emphasis on causal relationships. The Causal-TimeBank is explicitly annotated with causal links between events, making it a rich resource for training algorithms to recognize such patterns. Similarly, RFIs often contain causal queries related to construction processes or project management issues, albeit in an implicit manner.

```
</Markables>
<Relations>
  <TLINK comment="" id="1" relType="IS_INCLUDED">
    <source id="1" />
    <target id="0" />
  </TLINK>
  <TLINK comment="" id="2" relType="BEFORE">
    <source id="3" />
    <target id="4" />
  </TLINK>
  <TLINK comment="" id="3" relType="ENDED_BY">
    <source id="5" />
    <target id="0" />
  </TLINK>
  <TLINK comment="" id="4" relType="INCLUDES">
    <source id="5" />
    <target id="6" />
  </TLINK>
  <TLINK comment="" id="5" relType="INCLUDES">
    <source id="5" />
    <target id="8" />
  </TLINK>
  <TLINK comment="" id="6" relType="IS_INCLUDED">
    <source id="7" />
    <target id="5" />
  </TLINK>
  <TLINK comment="" id="7" relType="BEFORE">
    <source id="9" />
    <target id="0" />
  </TLINK>
```


Figure 4.4: Annotated relationship in CTB dataset.

Complex Sentence Constructions: Both datasets utilize complex sentence structures that often embody multiple clauses with causal or temporal connections. This complexity is typical in formal communications where detailed explanations or specifications are necessary, providing a good training ground for models to learn from sophisticated linguistic constructs. Formal Language and Standard Grammar: The use of standard grammatical constructions in both datasets ensures that a model trained on one is likely adaptable to the other without significant loss in accuracy due to linguistic variations. This standardization supports better generalization of the learned patterns across different text types.

Temporal References and Sequencing: Temporal elements in both datasets are crucial, as they often indicate the sequence of events or the timing of information needs. In RFIs, these might relate to project timelines or deadlines, whereas in Causal-TimeBank, they pertain to the timing and sequence of news events.

Given these similarities, a model developed or trained using the Causal-TimeBank dataset could

```xml
<Relations>
   <TLINK comment="" id="1" relType="IS_INCLUDED">
     <source id="2" />
     <target id="0" />
   </TLINK>
   <TLINK comment="" id="2" relType="BEFORE">
     <source id="3" />
     <target id="0" />
   </TLINK>
   <TLINK comment="" id="3" relType="BEFORE">
     <source id="4" />
     <target id="0" />
   </TLINK>
   <CLINK c-signalID="10" comment="" id="4">
     <source id="12" />
     <target id="11" />
   </CLINK>
   <CLINK comment="" id="5">
     <source id="13" />
     <target id="11" />
   </CLINK>
```

Figure 4.5: Annotated relationship in RFI dataset.

potentially be adapted to process and analyze RFIs. This cross-utilization can enhance the model's ability to discern and predict causal relationships within the highly structured and formalized text of RFIs, thereby aiding in more efficient management of information queries in construction projects. Leveraging the Causal-TimeBank's detailed annotations of causal links can also provide a foundational understanding that enhances the model's capability to handle the implicit causality often present in RFI queries. RFIs typically include cause-and-effect questions within construction projects, implying a need for recognizing and processing causal relationships within the inquiries and responses. Causal-TimeBank explicitly annotated for causal relationships between events. This dataset provides a structured approach to identify and classify causal links, making it a valuable resource for training models to detect and understand causal relationships in text.

4.1.6 Experimental Design and Hyperparameters

To further optimize the model's performance, fine-tuning was conducted after the initial experimentation phase. This iterative process involved systematically adjusting hyperparameters based on observed trends in performance metrics, focusing on configurations that enhanced precision and recall without compromising the F1 Score. The goal was to refine the model's ability to generalize across different graph structures while maintaining robustness under varying levels of input noise.

A total of 72 controlled experiments were conducted using different combinations of hyperparameters to evaluate their impact on model performance. Key parameters analyzed included the learning rate (`lr`), discount factor (`df`), exploration rate (`exr`), and exploration decay (`exd`). By varying these parameters systematically, we identified configurations that contributed to improved convergence rates, stability, and predictive accuracy.

Learning Rate (`lr`):

The learning rate determines the extent to which the model incorporates new information when updating Q-values. A high learning rate allows the model to adapt quickly to new data but may lead to instability or overfitting, as it rapidly discards previously learned values. Conversely, a low learning rate ensures more stable updates but can slow down convergence.

Tested values: 0.1, 0.3, 0.5, 0.7

Findings: An intermediate value (around 0.5) provided a balance between learning speed and stability.

Discount Factor (`df`):

The discount factor controls how much future rewards contribute to the current decision-making process. A high discount factor encourages the model to focus on long-term gains, which is beneficial for complex decision-making where delayed rewards matter. A low discount factor () prioritizes immediate rewards, which may lead to short-sighted decisions.

Tested values: 0.5, 0.7, 0.9, 0.95

Findings: Higher values (0.9 - 0.95) led to better performance, as the model effectively learned long-term causal and temporal dependencies in event graphs.

Exploration Rate (`exr`):

The exploration rate determines the balance between exploitation (choosing the best-known action) and exploration (trying new actions to discover better strategies). A high exploration rate () encourages the model to experiment more, reducing the risk of getting stuck in suboptimal policies. However, excessive exploration may lead to unstable performance, as the model may not settle on a well-learned policy.

Tested values: 0.3, 0.5, 0.7, 0.95

Findings: An initial high exploration rate (0.95) followed by gradual decay resulted in better convergence and generalization.

Exploration Decay (`exd`):

Exploration decay regulates how quickly the model shifts from an exploration-driven approach to a more exploitation-based strategy. If decay is too fast, the model may stop exploring before it learns the best policy. If it is too slow, the model may continue to explore unnecessarily, preventing convergence.

Tested values: 0.99, 0.995, 0.999

Findings: A decay factor of 0.995 provided a balanced transition between exploration and exploitation, allowing the model to explore initially and then refine its learned policy effectively.

A gradually decreasing exploration rate led to faster convergence and more accurate event link predictions. The balance between learning rate and discount factor significantly affected stability, ensuring the model did not oscillate between poor predictions.

Through these 72 controlled experiments, we were able to systematically refine the model and identify the most effective parameter combinations for improving graph refinement, causal/temporal inference, and predictive accuracy in RFI event graphs.

The learning rate was adjusted to determine the optimal step size for weight updates, balancing fast convergence and stability. A discount factor analysis was performed to examine its effect on balancing short-term and long-term rewards, ensuring that the reinforcement learning agent made decisions that optimized long-term outcomes. Additionally, exploration parameters were tuned to control the trade-off between exploration and exploitation. The initial exploration rate was set high to allow the model to search broadly across potential solutions, while the exploration decay was introduced to ensure a gradual shift toward exploiting learned policies as training progressed.

The results from these experiments informed the selection of the final hyperparameter values, ensuring an optimal balance between adaptability and precision. The final hyperparameters used in our method, determined after extensive fine-tuning, are as follows: the learning rate is set to 0.5, enabling efficient updates while avoiding instability; the discount factor is set to 0.95, ensuring a

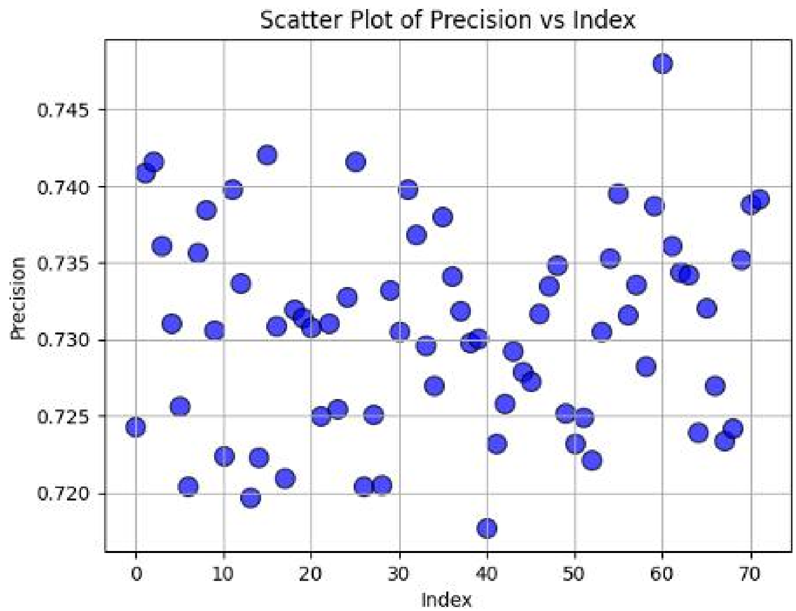

Figure 4.6: Scatter plot of Precision in different hyperparameters.

strong consideration of future rewards over immediate gains. The initial exploration rate starts at 0.99, allowing the agent to thoroughly explore possible configurations early in training, with an exploration decay factor of 0.98 to gradually transition from exploration to exploitation as learning progresses. The model underwent training for 200, 500, and 1000 iterations, with the exploration rate dynamically decreasing over time to ensure a smooth convergence toward optimal graph refinement.

This experimental setup provided valuable insights into the effects of hyperparameter tuning on reinforcement learning performance. By leveraging a structured experimental design, we ensured that the model effectively adapted to varying error levels in input graphs while maintaining high generalization capabilities. The final configurations significantly improved the model's robustness, making it more capable of handling complex causal and temporal relationships with greater accuracy.

4.1.7 Evaluation Metrics

To evaluate the quality of the refined graph G' relative to the ground truth graph G_{true}, we employ three key performance metrics: Precision, Recall, and F1 Score. These metrics provide a

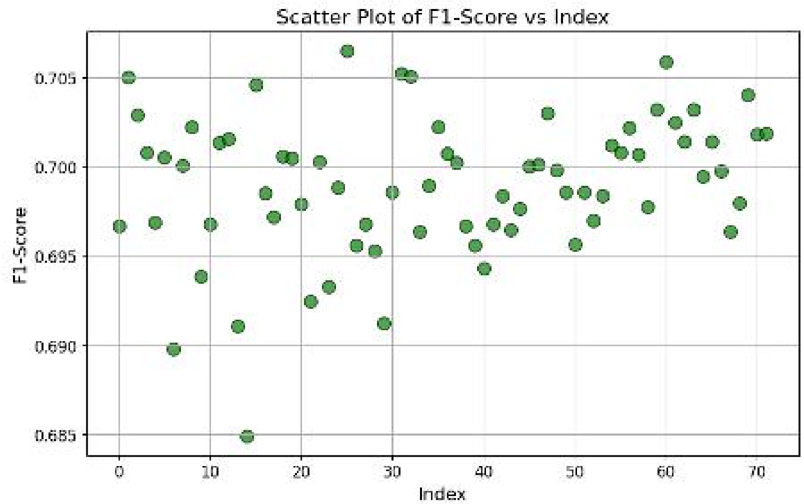

Figure 4.7: Scatter plot of F1 score in different hyperparameters.

comprehensive assessment of how well the reinforcement learning model refines the initial graph G_{init} and ensures that the generated causal and temporal relationships align with expert-annotated ground truth data.

Precision measures the proportion of correctly identified edges in the final graph G_{final} relative to all predicted edges. A high precision value indicates that the model effectively reduces false positives, minimizing the introduction of spurious links that do not exist in G_{true}. Precision is computed as follows:

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{14}$$

where TP (True Positives) represents correctly predicted edges that exist in both G_{final} and G_{true}, and FP (False Positives) denotes incorrectly added edges that are present in G_{final} but absent from G_{true}.

Recall evaluates the model's ability to recover all correct edges from G_{true}. A high recall value signifies that the model successfully retains essential causal and temporal relationships without omitting important connections. Recall is computed as:

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{15}$$

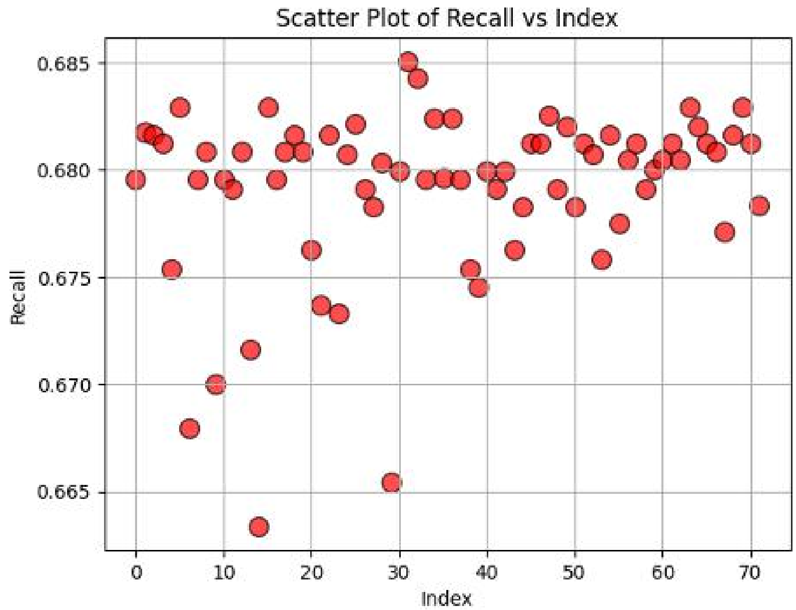

Figure 4.8: Scatter plot of Recall in different hyperparameters.

where FN (False Negatives) refers to edges that are missing from G_{final} but exist in G_{true}. A low recall value indicates that the model may be overly conservative, failing to detect critical relationships.

To balance the trade-off between Precision and Recall, we compute the F1 Score, which provides a harmonic mean of both metrics. The F1 Score ensures that the model optimizes both precision (avoiding incorrect edges) and recall (ensuring necessary edges are included). It is given by:

$$\text{F1 Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{16}$$

A high F1 Score signifies that the model achieves a strong balance between reducing false positives and avoiding false negatives, thereby improving the reliability of the refined graph.

By incorporating these three evaluation metrics, we can systematically assess the effectiveness of the reinforcement learning framework in refining causal-temporal graphs. Additionally, these metrics provide a quantitative measure of how well the model generalizes across different datasets, ensuring that it consistently produces high-quality graph structures in various real-world scenarios.

Table 4.2: Confusion Matrix for Graph Evaluation

	Predicted Edge Exists	**Predicted Edge Absent**
True Edge Exists	True Positive (TP)	False Negative (FN)
True Edge Absent	False Positive (FP)	True Negative (TN)

4.2 Implementation and Performance Evaluation

This section provides a comprehensive evaluation of the performance of our Hierarchical Reinforcement Learning with Human Feedback (HRLHF) model on the Causal-TimeBank dataset. The evaluation focuses on assessing the model's ability to iteratively refine causal and temporal relationships, ensuring the generated graph closely aligns with the expert-annotated ground truth. To gain deeper insights into the model's learning process, we visualize cumulative rewards and track reward convergence throughout training, enabling a better understanding of how the model optimizes its decision-making over time.

4.2.1 Initial Graph Generation with Simulating Human-Generated Variants

The primary objective of the experiment is to evaluate the model's capability to transform an initial error-induced graph into a more accurate representation of the true graph structure. The initial graph, containing synthetic errors, represents a noisy version of the ground truth, simulating real-world uncertainties in causal and temporal inference. By comparing the initial graph with the final refined graph produced after applying the HRLHF framework, we measure the extent to which the model effectively reduces errors and improves structural accuracy.

To generate realistic noisy datasets, two primary types of errors were systematically introduced into the initial graph:

- **Misprediction Errors:** These involve the random addition or removal of edges between nodes in the graph. Incorrect edge modifications may distort causal dependencies, leading to incorrect inferences about event relationships.

- **Misclassification Errors:** These occur when the type of an existing edge is altered, potentially misrepresenting causal and temporal link classifications. This form of error can disrupt the logical sequencing of events, affecting interpretability.

By introducing these controlled errors, we simulate realistic challenges encountered in automated causal inference tasks. The ability of HRLHF to recover from these errors and refine the graph structure serves as a key indicator of its robustness. Through systematic evaluation, we observe that HRLHF effectively corrects erroneous relationships while retaining valid ones, demonstrating its ability to enhance causal and temporal modeling accuracy in real-world applications.

4.2.2 Quantitative Performance Metrics

Our evaluation involves a quantitative comparison of precision, recall, and F1 Score to assess the improvements achieved by HRLHF. Additionally, we analyze the impact of different error rates on model performance by introducing varying degrees of misprediction and misclassification errors into the initial graph. The effectiveness of HRLHF is determined by the extent to which it corrects these errors while maintaining a balance between exploration and exploitation during training.

The evaluation results are presented in Tables 4.3 and 4.4, which compare the performance of the initial graph containing errors before HRLHF optimization (Table 4.3) to that of the final graph after HRLHF optimization (Table 4.4). To ensure statistical reliability and generalizability, the reported performance metrics were averaged across 30 experimental trials. Each experiment was conducted using graphs with varying levels of artificially introduced errors, enabling a robust assessment of the framework's effectiveness under different noise conditions. The results consistently demonstrate that HRLHF significantly enhances graph accuracy across a broad range of error scenarios.

Graphs with error rates of 5%, 10%, 20%, and 40% were used to evaluate the robustness of the model. For the misprediction error, which includes adding or removing edges, the initial F1 Score decreased significantly as error rates increased, from 0.86 at a 5% error rate to 0.60 at a 40% error rate. However, after applying HRLHF, the F1 Score improved across all error rates, reaching 0.91 at 5% error and 0.64 at 40% error, respectively. Similarly, for the misclassification error type, which involves modifying the edge type of relationships, the initial F1 Score dropped from 0.89 at 5%

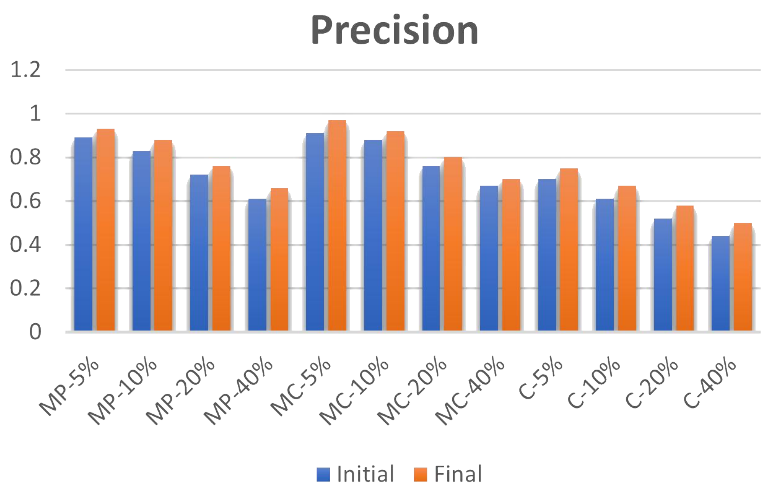

Figure 4.9: Precision before and after RL.

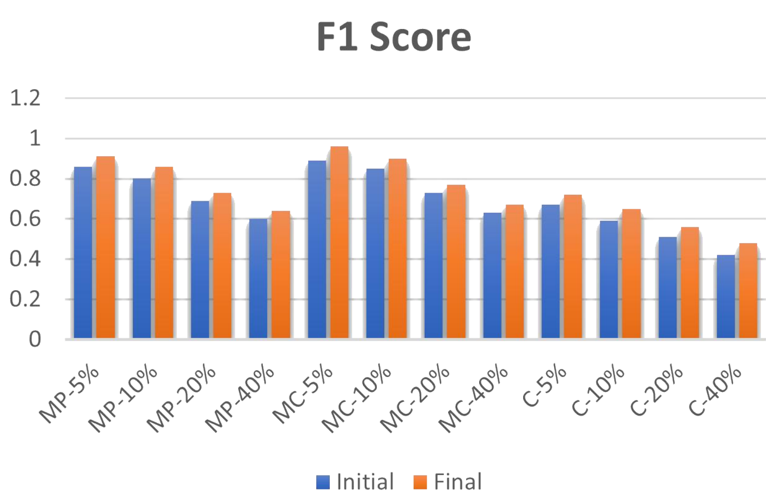

Figure 4.10: F1-Score before and after RL.

Table 4.3: Initial Evaluation

Error Type	**Error Rate**	**Precision**	**Recall**	**F1 Score**
Link Misprediction	5%	0.89	0.85	0.86
	10%	0.83	0.78	0.80
	20%	0.72	0.66	0.69
	40%	0.61	0.58	0.60
Link Misclassification	5%	0.91	0.87	0.89
	10%	0.88	0.83	0.85
	20%	0.76	0.71	0.73
	40%	0.67	0.60	0.63
Combined	5%	0.70	0.65	0.67
	10%	0.61	0.58	0.59
	20%	0.52	0.50	0.51
	40%	0.44	0.40	0.42

Table 4.4: Final Evaluation

Error Type	**Error Rate**	**Precision**	**Recall**	**F1 Score**
Link Misprediction	5%	0.93	0.90	0.91
	10%	0.88	0.85	0.86
	20%	0.76	0.71	0.73
	40%	0.66	0.63	0.64
Link Misclassification	5%	0.97	0.95	0.96
	10%	0.92	0.89	0.90
	20%	0.80	0.75	0.77
	40%	0.70	0.65	0.67
Combined	5%	0.75	0.70	0.72
	10%	0.67	0.64	0.65
	20%	0.58	0.55	0.56
	40%	0.50	0.46	0.48

error to 0.63 at 40% error, respectively. After optimization, the F1 Score increased to 0.96 at 5% error and 0.67 at 40% error, reflecting the framework's strong ability to correct semantic errors.

HRLHF demonstrated greater robustness in correcting misclassification errors than misprediction errors, as evidenced by consistently higher F1 Scores across all error rates. The framework achieved near-perfect optimization at low error rates (5%), with F1 Scores exceeding 0.91 for both error types. At high error rates (40%), while performance naturally declined due to substantial error, HRLHF still improved graph accuracy by 6–9%, showcasing its ability to recover meaningful relationships. The high-level policy demonstrates 86.12% accuracy, effectively deciding when to add or remove links, ensuring a well-structured causal graph. Similarly, the low-level policy achieves 84.87% accuracy, indicating strong performance in accurately classifying CLINK and TLINK relationships after the high-level actions.

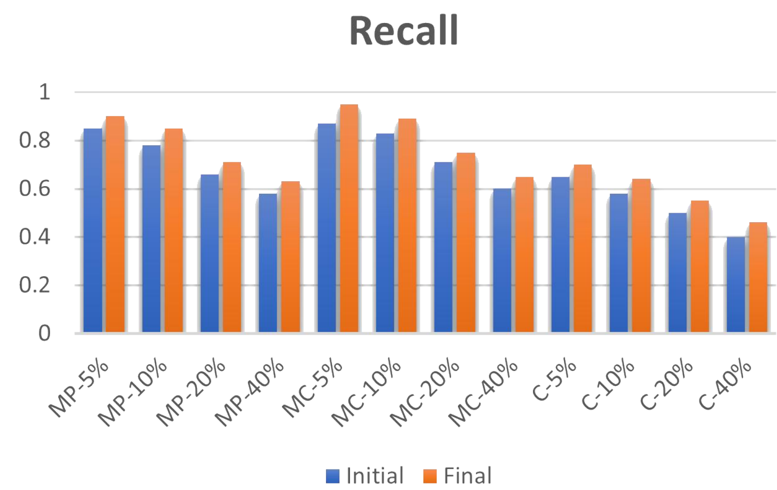

Figure 4.11: Recall before and after RL..

Figure 4.12 illustrates the progression of immediate reward per iteration during the training of the HRLHF model. As seen in the plot, the reward begins at a low baseline—frequently dipping below zero in early iterations—indicating initial instability or incorrect graph predictions. However, as training progresses, the reward values show a clear stepwise increase, reflecting the model's learning curve. These discrete jumps suggest the agent is successfully refining its policy over time, with reinforcement feedback leading to more accurate causal and temporal link predictions. Notably, the rewards stabilize around 1000 after approximately 500–600 iterations, showing that the agent converges toward an effective strategy. This pattern confirms that the reward function is well-structured and capable of guiding the agent toward optimal graph configurations.

Figure 4.13 displays the convergence of the maximum Q-value over time during training. Initially, the Q-values fluctuate significantly, with high variance and sharp spikes both above and below zero. This indicates that the agent is still exploring and has not yet developed a stable policy. Around iteration 400 to 500, there is a notable peak, suggesting a breakthrough in learning where the agent identifies high-value actions. After this phase, the fluctuations gradually decrease, and the Q-values begin to stabilize, especially beyond iteration 600. This trend reflects the transition from exploration to exploitation, where the agent refines its decisions based on accumulated learning. The convergence pattern confirms that the model's Q-learning process is functioning correctly, progressively reducing uncertainty and honing in on more reliable action-value estimates.

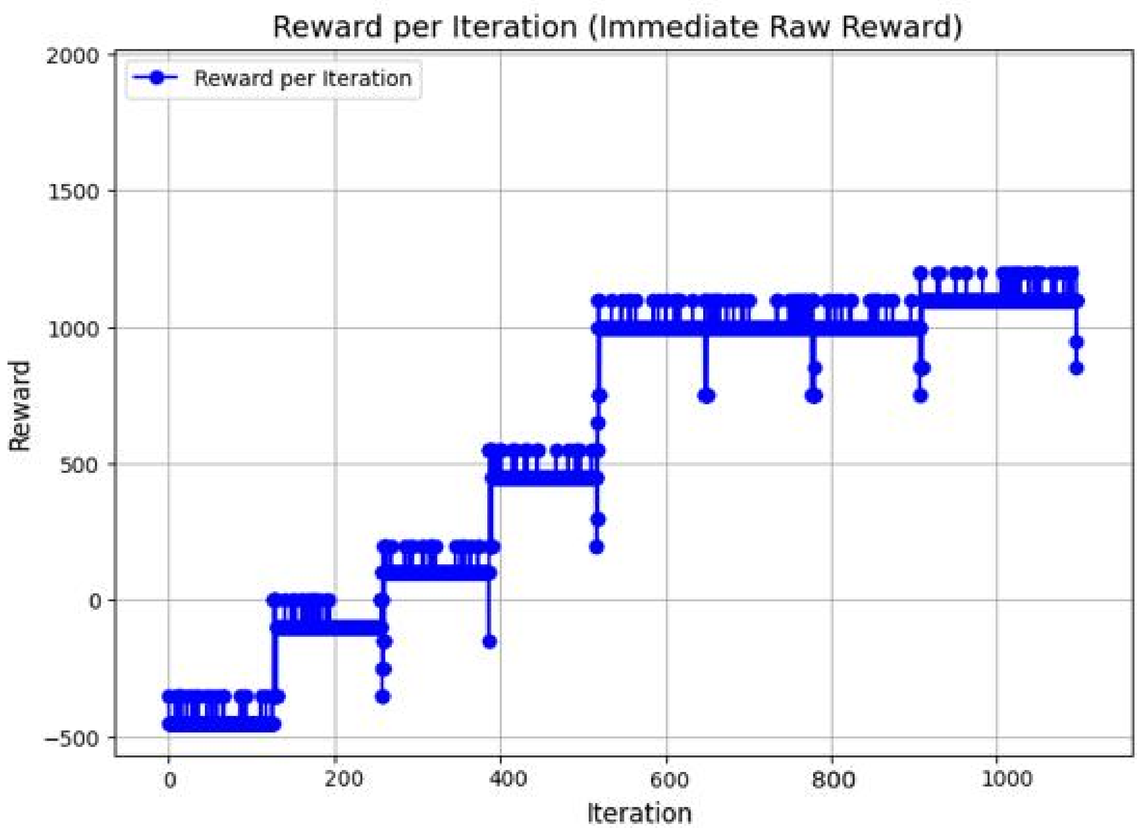

Figure 4.12: Immediate reward through the training process, which converges after 500 iterations.

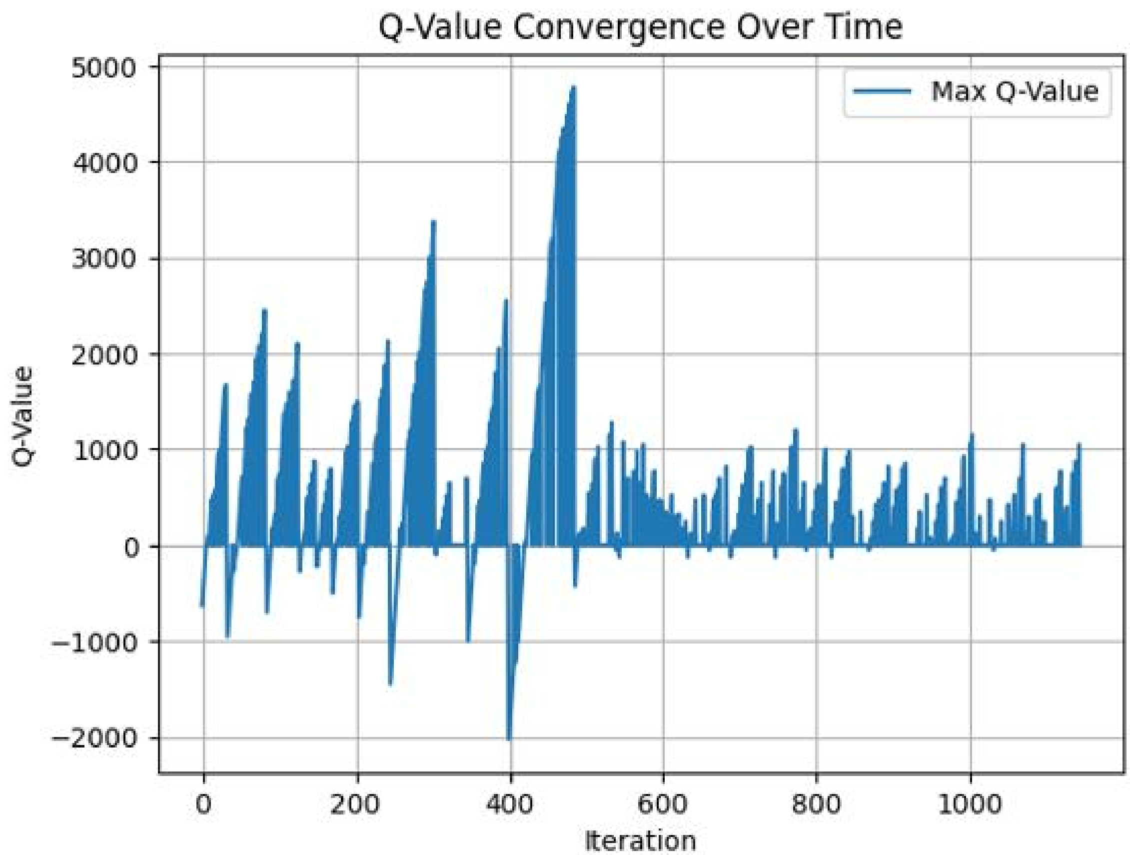

Figure 4.13: Maximum Q-value across all state-action pairs at each iteration, which is stable after 500 iterations.

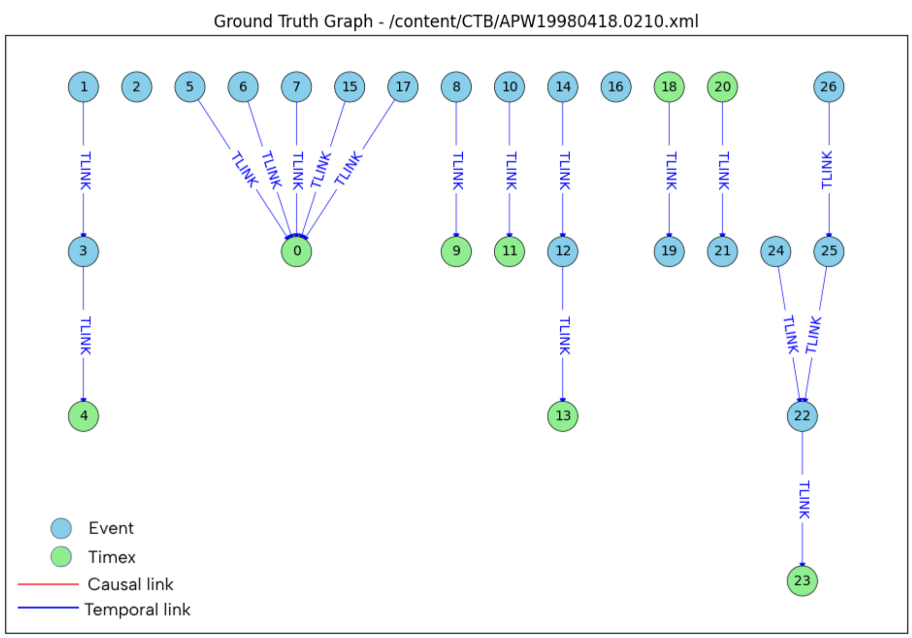

Figure 4.14: Ground truth graph that is generated from the true dataset.

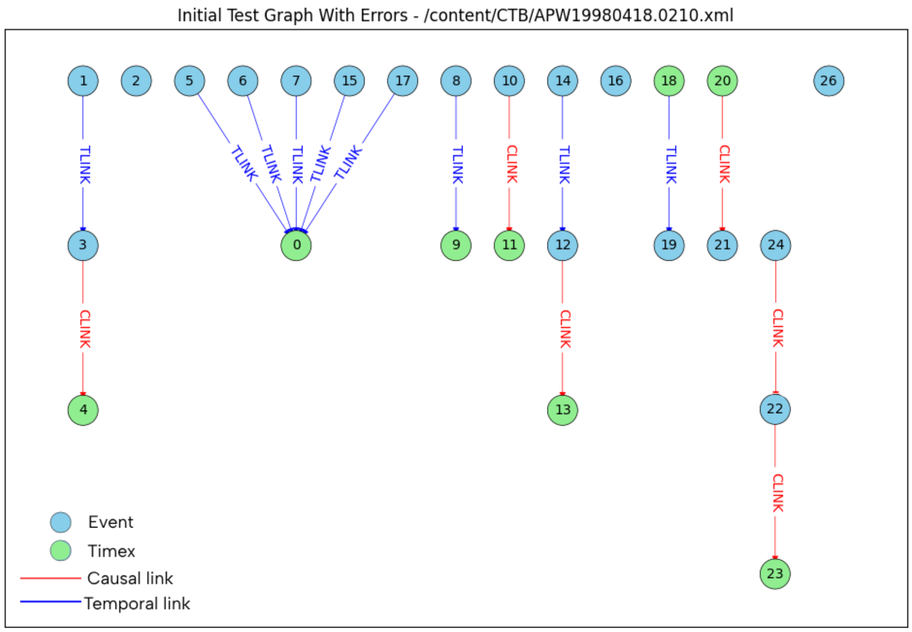

Figure 4.15: Initial graph generated from the dataset with some errors.

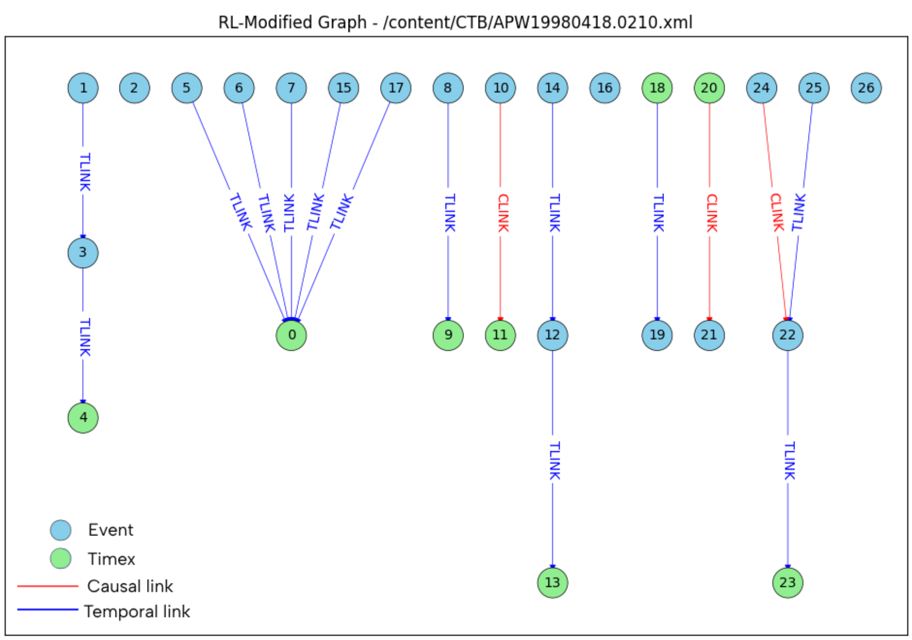

Figure 4.16: Modified graph after HRLHF.

By leveraging reinforcement learning, the model progressively learns optimal policies for graph refinement, guided by a hierarchical decision-making framework. The convergence of cumulative rewards throughout training serves as an indicator of the model's stability and efficiency in improving causal and temporal link predictions. A well-converged reward function suggests that the model has successfully learned to minimize incorrect modifications while preserving correct relationships, leading to a more accurate graph representation.

Overall, this evaluation provides valuable insights into the practical applicability of HRLHF in refining causal-temporal graphs. The results demonstrate how reinforcement learning, when combined with human feedback, enhances the interpretability and accuracy of causal inferences, making it a powerful tool for analyzing event dependencies in structured datasets.

4.2.3 Causal and Temporal Link Evaluation

Table 4.5 compares CLINK (causal links) and TLINK (temporal links) statistics across some 10 random files and different stages, ground truth (CL/TL-GT), initial graph (CL/TL-IG), and after RL (CL/TL-RL). The results indicate that the model performs better on TLINK errors than CLINK errors, as TLINK counts consistently improve across all files. This suggests that temporal relationships are easier to learn and correct, whereas causal dependencies require deeper reasoning and remain more challenging for the model.

Table 4.5: CLINK and TLINK statistics, including correct initial and final links

File #	CL-GT	CL-IG	CL-RL	TL-GT	TL-IG	TL-RL
1	2	1	2	25	12	20
2	2	0	1	20	12	19
3	2	0	2	6	4	6
4	5	1	3	52	32	49
5	10	4	7	50	30	49
6	6	0	3	8	5	8
7	5	3	4	35	19	31
8	6	2	4	70	42	68
9	4	0	2	9	5	8
10	2	1	2	29	16	27

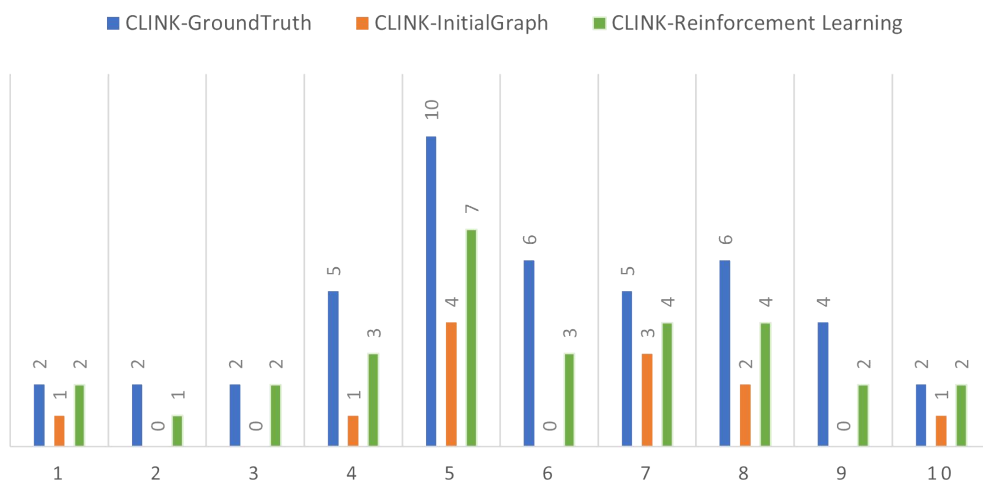

Figure 4.17: CLINK Evaluation.

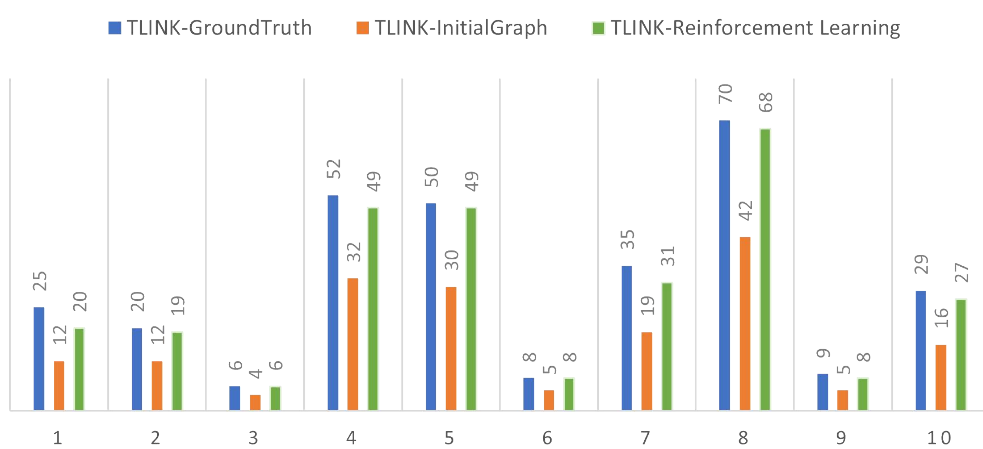

Figure 4.18: TLINK Evaluation.

4.2.4 Statistical Evaluation

The statistical test confirms a significant improvement in Precision, Recall, and F1 Score. The paired t-tests show that the final evaluation outperforms the initial evaluation with t-statistics of -16.26, -17.85, and -17.68 for Precision, Recall, and F1 Score, respectively, all with p-values below 10^{-9}. This strong statistical significance indicates that the RL-based approach effectively refines the graph structure. These results validate the model's ability to mitigate errors and enhance link prediction quality, demonstrating its robustness in structured learning tasks.

Table 4.6: Statistical Test Results for Initial vs. Final Evaluation

Metric	**t-test**	**p-value**
Precision	-16.26	4.87×10^{-9}
Recall	-17.85	1.80×10^{-9}
F1 Score	-17.68	2.00×10^{-9}

The F1 Score and precision improvements result from human feedback (HF), which helps the model refine predictions iteratively. Figure 4.19 illustrates the HRL model's learning progression, with cumulative rewards over 1,000 iterations. The cyan line shows fluctuating rewards, with an overall upward trend indicating performance improvement. The thick blue line, a moving average, smooths fluctuations and steadily increases, confirming learning progress. The shaded area represents variance, showing stabilization, while a slight dip near the end suggests momentary exploration of a suboptimal strategy. Overall, the graph demonstrates that the model is improving steadily through exploration and refinement. In summary, combining HRL with HF enables the model to learn from data while refining its predictions through expert input. This enhances accuracy and reliability, as seen in the higher F1 scores and cumulative rewards.

4.3 Summary

The experimental evaluation of the HRLHF model was conducted using the Causal-TimeBank dataset, which consists of annotated causal and temporal relationships between events. The dataset provided a structured environment for testing the model's ability to refine event graphs and improve their accuracy. To assess the model's robustness, errors were introduced into the graphs, simulating

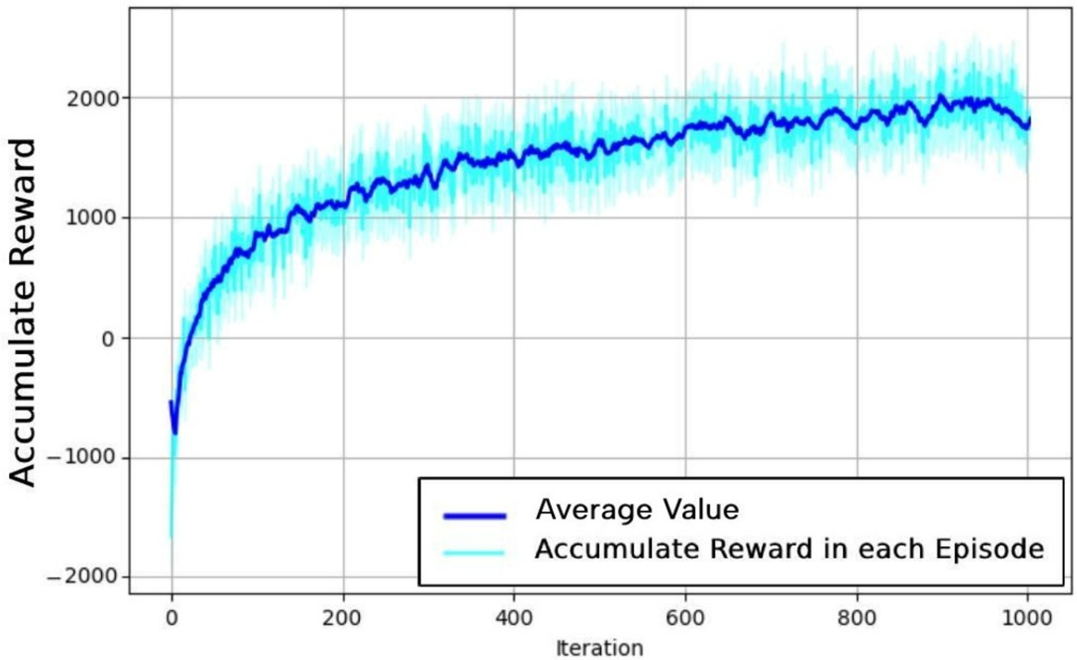

Figure 4.19: Accumulated rewards over 1000 iterations to show the RL performance progression.

real-world uncertainties. These errors included both misprediction errors (incorrect additions or removals of edges) and misclassification errors (incorrectly labeling edge types as causal or temporal).

The model's performance was analyzed using precision, recall, and F1-score, ensuring a comprehensive evaluation of its capability to refine noisy event graphs. Results demonstrated that HRLHF significantly improved the accuracy of causal and temporal link classifications, with F1-scores showing substantial improvements across different error rates. The reward function convergence analysis indicated that the model effectively learned optimal graph structures, with reward stabilization occurring after multiple iterations.

In addition, the study compared the effectiveness of the high-level policy (which determines whether to add or remove edges) and the low-level policy (which refines the classification of edges). Findings revealed that temporal relationships (TLINKs) were easier to correct than causal relationships (CLINKs), highlighting the model's strengths and challenges.

Overall, the experimental results validated the effectiveness of HRLHF in refining event graphs for RFI analysis, demonstrating its potential application in real-world scenarios where structured learning is required for causal and temporal inference.

Chapter 5

Discussion

5.1 Interpretation of Findings

The study presented several key observations that were not explicitly concluded in the main results. Some unexpected trends emerged, including variations in the annotation consistency of RFIs compared to the Causal-TimeBank dataset. While the overall structure of RFIs aligns with causal event annotations, certain complexities arose in defining implicit causal relationships within project-specific documents. These patterns highlight the necessity for refining annotation techniques to improve automated processing.

Comparing the findings with prior research, the alignment of RFIs with event-based causal annotation frameworks indicates potential for expanding annotation schemes. However, domain-specific adaptations may be required to ensure accuracy in capturing nuanced causal dependencies unique to construction-related RFIs.

5.2 Limitations and Challenges

Several methodological constraints were encountered during this study. One key limitation was the availability and quality of annotated RFI datasets, which required additional preprocessing steps. Inconsistencies in annotation between different annotators led to variations in identifying temporal and causal links, emphasizing the need for stricter annotation guidelines.

Technical challenges included computational constraints in processing large-scale document corpora. The use of automated annotation tools such as BRAT and GATE provided efficiency, but the reliance on manual intervention for validation introduced potential biases. Additionally, the lack of standard benchmarks for evaluating RFI causal annotations remains a limitation.

5.3 Practical and Theoretical Implications

The findings contribute to both practical and theoretical advancements in automated document annotation. From a practical standpoint, the proposed annotation methodology can enhance project management workflows by improving the efficiency of RFI handling through NLP-driven automation. The identification of causal and temporal dependencies in RFIs can assist construction teams in tracking project developments more effectively.

Theoretically, this study extends the application of causal annotation frameworks beyond news texts and general datasets. By demonstrating the feasibility of RFI annotation using the Causal-TimeBank structure, this research lays the groundwork for future NLP applications in domain-specific document processing.

5.4 Future Directions

Future research should explore refining annotation frameworks to enhance consistency in identifying causal relationships in RFIs. Potential extensions include:

- Incorporating deep learning models for improved causal inference in textual documents.
- Expanding the annotation schema to accommodate more nuanced relationships in construction RFIs.
- Applying similar annotation techniques to other domain-specific documents, such as legal or medical records.

- Developing automated benchmarking tools to assess annotation quality and consistency.

These directions will further validate the applicability of automated RFI analysis and its integration into real-world project management systems.

5.5 Unanswered Questions

While the results of this study demonstrate the potential of hierarchical reinforcement learning with human feedback (HRLHF) in refining event graphs for Request for Information (RFI) analysis, several open questions remain that warrant further investigation. These unanswered questions highlight both methodological and practical challenges that could shape future research in this domain.

One important area for future investigation is the identification of implicit causal relationships within RFIs. While the model has excelled at capturing explicit causal and temporal links, RFIs often contain indirect causality that may not be easily detectable with current methods. How can future advancements in natural language processing (NLP) and graph-based learning help uncover these hidden relationships? Would integrating external knowledge graphs or pre-trained language models enhance the system's ability to detect deeper causal patterns?

Another critical question revolves around the balance between human feedback and automation in model refinement. The study showed that expert feedback significantly improved the accuracy of link classification, but the ideal level of human intervention remains an open question. How can semi-supervised learning or active learning strategies be leveraged to optimize the trade-off between human expertise and model autonomy? Can interactive learning frameworks further streamline the annotation process without sacrificing quality?

Scalability is also a crucial consideration. The model performed well on the tested dataset, but how effectively can it generalize to different types of construction projects, industries, and structured documents? Exploring domain adaptation techniques will be essential for extending HRLHF's application to a broader range of datasets. Can transfer learning approaches facilitate seamless adaptation of the model to different RFI formats and contexts?

Addressing these unanswered questions will be crucial for refining annotation methodologies, improving model generalization, and enhancing the reliability of AI-driven causal analysis for structured document processing. Future research in these areas will not only strengthen the effectiveness of HRLHF in RFI analysis but also contribute to broader applications in automated causal inference and event-based decision support systems.

Chapter 6

Conclusion and Future Work

This paper proposes an HRLHF model to improve the identification of causal and temporal relationships in the Causal-TimeBank data set. Although the experiments were conducted using the Causal-TimeBank dataset, a general-purpose, well-structured benchmark for causal-temporal inference, our contribution explicitly targets the challenge of graph refinement based on structured textual event data. This challenge is directly relevant to the construction of RFIs, which often contain rich textual descriptions of project interactions that can be annotated with causal and temporal relationships. Although this work does not use real RFI datasets, it demonstrates a scalable method for refining event graphs under the assumption that events are pre-extracted and structured. The model shows a strong performance in recovering accurate causal and temporal links from partially erroneous graphs.

To validate the effectiveness of our HRL framework and HF integration, we present case studies analyzing performance improvements, error breakdowns by edge types, and practical implications. Our evaluation methodology involves comparing initial error-induced graphs to their refined counterparts after HRLHF optimization. We assess how hierarchical decision-making and human feedback contribute to refining causal and temporal graphs, emphasizing how human intervention assists in correcting missing and inaccurate edges. The results indicate that HRL enhances structural learning by progressively improving causal and temporal link predictions, while human feedback accelerates error correction and boosts model generalization. Precision, recall, and F1 Score metrics provide quantitative validation of the improvements achieved through our approach.

By integrating domain expertise into automated learning, our HRLHF framework enhances graph accuracy and interpretability, making it applicable to construction data analysis, process mining, and other domains requiring structured event-based decision-making. This framework provides a scalable mechanism for addressing noisy and ambiguous relationships in large datasets, offering robust solutions for understanding complex event interactions. Additionally, the model's ability to incorporate human oversight allows for adaptive learning, making it suitable for dynamically evolving datasets where new event dependencies emerge over time.

Future work will focus on scaling the model to domain-specific datasets, assessing its performance on complex, diverse data, and extending its applicability to various real-world scenarios. Specifically, we aim to adapt the HRLHF framework for use in construction project management, where RFIs, change orders, and progress reports involve intricate causal-temporal dependencies. Moreover, the approach can be evaluated in healthcare, where patient histories and treatment plans require accurate cause-effect modeling, financial systems, where fraud detection and market trend analysis rely on event-based inference, and natural disaster analysis, where forecasting relies on understanding environmental and human-driven interactions. Expanding the model's applicability across these domains will further validate its robustness, adaptability, and potential impact in advancing causal and temporal inference methodologies.

References

Afzal, M., Wong, J., and Fini, A. (2024). Towards digital approach for managing request for information (rfi) in construction projects: a literature review. *Construction Innovation*.

Afzal, M., Wong, J. K. W., and Fini, A. A. F. (2023). Unlocking insights: Analysing construction issues in request for information (rfi) documents with text mining and visualisation. In *2023 IEEE 19th International Conference on Automation Science and Engineering (CASE)*, pages 1–6. IEEE.

Aibinu, A. A., Carter, S., Francis, V., and Vaz-Serra, P. (2020). Request for information frequency and their turnaround time in construction projects: A data-analytic study. *Built environment project and asset management*, 10(1):1–15.

Alrasheed, K., Soliman, E., and Albader, H. (2023). Delay dispute cases: Comparative analysis and claimed value prediction model. *Journal of Legal Affairs and Dispute Resolution in Engineering and Construction*, 16.

Asghari, V., Wang, Y., Jahanbiglari, A., Hsu, S.-C., and Tang, P. (2022). Reinforcement learning in construction engineering and management: A review. *Journal of Construction Engineering and Management*, 148.

Bai, Y., Jones, A., Ndousse, K., Askell, A., Chen, A., DasSarma, N., Drain, D., Fort, S., Ganguli, D., Henighan, T., et al. (2022). Training a helpful and harmless assistant with rlhf. *arXiv preprint arXiv:2204.05862*.

Bai, Z., Wang, P., Xiao, T., He, T., Han, Z., Zhang, Z., and Shou, M. Z. (2024). Hallucination of multimodal large language models: A survey. *ArXiv*, abs/2404.18930.

Bhat, A. S. (2017). *Data visualization of requests for information to support construction decision-making*. PhD thesis, University of British Columbia.

Bignold, A., Cruz, F., Dazeley, R., Vamplew, P., and Foale, C. (2021). An evaluation methodology for interactive reinforcement learning with simulated users. *Biomimetics*, 6(1):13.

Caselli, T., Lenzi, V. B., Sprugnoli, R., Pianta, E., and Prodanof, I. (2011). Annotating events, temporal expressions and relations in italian: the it-timeml experience for the ita-timebank. In *Proceedings of the 5th linguistic annotation workshop*, pages 143–151.

Chen, Y. et al. (2023). Integrating causal inference into data-driven parametric analysis in aec. *ArXiv*, abs/2309.11509.

Gopalakrishnan, S., Chen, V. Z., Dou, W., Hahn-Powell, G., Nedunuri, S., and Zadrozny, W. (2023). Text to causal knowledge graph: A framework to synthesize knowledge from unstructured business texts into causal graphs. *Information*, 14(7):367.

Hanna, A. S., Tadt, E. J., and Whited, G. C. (2012). Request for information: Benchmarks and metrics for major highway projects. *Journal of Construction Engineering and Management*, 138(12):1347–1352.

Hughes, N., Wells, M., Nutter, C., and Zack, J. (2013). Impact & control of rfis on construction projects. *Navigant Construction ForumTM*.

Jallan, Y., Brogan, E., Ashuri, B., and Clevenger, C. M. (2019). Application of natural language processing and text mining to identify patterns in construction-defect litigation cases. *Journal of legal affairs and dispute resolution in engineering and construction*, 11(4):04519024.

Johnson, F. and Dana, K. (2023). Hierarchical reinforcement learning for temporal pattern prediction.

Kelly, D. and Ilozor, D. (2020). Performance outcome assessment of the integrated project delivery (ipd) method for commercial construction projects in usa. *International Journal of Construction Management*, 22:1–9.

Labs, S. (2024). The process of sending an rfi in construction: A guide. Accessed: 27-Feb-2025.

Langbridge, A., O'Donncha, F., Ba, A., Lorenzi, F., Lohse, C., and Ploennigs, J. (2023). Causal temporal graph convolutional neural networks (ctgcn).

Lenzi, V. B., Moretti, G., and Sprugnoli, R. (2012). Cat: the celct annotation tool. In *LREC*, pages 333–338.

Li, D., Yan, L., Zhang, X., Jia, W., and Ma, Z. (2023). Eventkge: Event knowledge graph embedding with event causal transfer. *Knowledge-based systems*, 278:110917.

Lin, Z.-X., Xu, Q., and Luo, Z.-W. (2020). A survey on interactive reinforcement learning from human social feedback. *Neural Networks*, 132:110–123.

Liu, K., Chen, Y., Liu, J., Zuo, X., and Zhao, J. (2020). Extracting events and their relations from texts: A survey on recent research progress and challenges. *AI Open*, 1:22–39.

Liu, X., Shao, Q., and Chen, D. (2024). Long-Term Prediction on Graph Data With Causal Network Construction . *IEEE Transactions on Artificial Intelligence*, 5(07):3445–3455.

Love, P. E., Zhou, J., Sing, C. P., and Kim, J. T. (2014). Assessing the impact of rfis in electrical and instrumentation engineering contracts. *Journal of Engineering Design*, 25(4-6):177–193.

Mirza, P. (2016). Extracting temporal and causal relations between events.

Mirza, P., Sprugnoli, R., Tonelli, S., and Speranza, M. (2014). Annotating causality in the tempeval-3 corpus. In *Proceedings of the EACL 2014 Workshop on Computational Approaches to Causality in Language (CAtoCL)*, pages 10–19, Gothenburg, Sweden.

Association for Computational Linguistics.

Mirza, P. and Tonelli, S. (2014). An analysis of causality between events and its relation to temporal information. In Tsujii, J. and Hajic, J., editors, *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 2097–2106, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Mohamed, S., Tilley, P., and Tucker, S. (1999). Quantifying the time and cost associated with the request for information (rfi) process in construction. *Construction Innovation: information, process, management*, 7(1):35–50.

Morales, F., Herrera, R. F., Rivera, F. M.-L., Atencio, E., and Nuñez, M. (2022). Potential application of bim in rfi in building projects. *Buildings*, 12(2).

Mosqueira-Rey, E., Hernández-Pereira, E., Alonso-Ríos, D., Bobes-Bascarán, J., and Fernández-Leal, Á. (2023). Human-in-the-loop machine learning: a state of the art. *Artificial Intelligence Review*, 56(4):3005–3054.

Ouyang, L., Wu, J., Jiang, X., Almeida, D., Wainwright, C., Mishkin, P., Zhang, C., Agarwal, S., Slama, K., Ray, A., et al. (2022). Training language models to follow instructions with human feedback. *Advances in Neural Information Processing Systems (NeurIPS)*.

Panahi, R., Kivlin, J. P., and Louis, J. (2023). Request for information (rfi) recommender system for pre-construction design review application using natural language processing, chat-gpt, and computer vision. In *Computing in Civil Engineering 2023*, pages 159–166.

Papajohn, D. and El Asmar, M. (2021). Impact of alternative delivery on the response time of requests for information for highway projects. *Journal of Management in Engineering*, 37(1):04020098.

Peng, S., Hu, X., Zhang, R., Tang, K., Guo, J., Yi, Q., Chen, R., Zhang, X., Du, Z., Li, L., Guo, Q., and Chen, Y. (2022). Causality-driven hierarchical structure discovery for reinforcement learning.

Philips-Ryder, M., Zuo, J., and Jin, X. H. (2013). Evaluating document quality in construction projects–subcontractors' perspective. *International Journal of Construction Management*, 13(3):77–94.

Pustejovsky, J., Hanks, P., Saurí, R., See, A., Gaizauskas, R., Setzer, A., Radev, D., Sundheim, B., Day, D., Ferro, L., and Lazo, M. (2003). The timebank corpus. *Proceedings of Corpus Linguistics*.

Pustejovsky, J., Knippen, R., Littman, J., and Saurí, R. (2005). Temporal and event information in natural language text. *Language resources and evaluation*, 39:123–164.

Retzlaff, C. O., Das, S., Wayllace, C., Mousavi, P., Afshari, M., Yang, T., Saranti, A., Angerschmid, A., Taylor, M. E., and Holzinger, A. (2024). Human-in-the-loop reinforcement learning: A survey and position on requirements, challenges, and opportunities. *Journal of Artificial Intelligence Research*, 79:359–415.

Shayesteh, S. and Jebelli, H. (2022). Toward human-in-the-loop construction robotics: Understanding workers' response through trust measurement during human-robot collaboration. In *Construction research congress 2022*, pages 631–639.

Shen, Y., Lin, J., Yang, X., Xu, J., and Wang, W. (2024). Improving reinforcement learning from human feedback with contrastive rewards. *Proceedings of the International Conference on Learning Representations (ICLR)*.

Shim, E., Carter, B., and Kim, S. (2016). Request for information (rfi) management: a case study. In *Proceedings of the 52nd ASC Annual International Conference Proceedings, Provo, UT, USA*, pages 13–16.

Shrestha, R., Ko, T., and Lee, J. (2023). Natural language processing (nlp)-driven classification of pre-bid request for information (rfi). In *Computing in Civil Engineering 2023*.

Smartsheet (2025). Construction rfi templates. Accessed: Feb. 27, 2025.

Soleymani, M., Bonyani, M., and Attarzadeh, M. (2022). Autonomous resource management in construction companies using deep reinforcement learning based on iot.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, second edition.

Thompson, P., Yates, T., Inan, E., and Ananiadou, S. (2020). Semantic annotation for improved safety in construction work. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 1990–1999.

Verhagen, M., Knippen, R., Mani, I., and Pustejovsky, J. (2006). Annotation of temporal relations with tango. In *LREC*, pages 2249–2252.

Wang, L., Zhang, C., Ding, R., Xu, Y., Chen, Q., Zou, W., et al. (2023). Root cause analysis for microservice systems via hierarchical reinforcement learning from human feedback. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5116–5125.

Xu, X. and Garc de Soto, B. (2022). Reinforcement learning with construction robots: A review of research areas, challenges and opportunities. In Linner, T., Garc de Soto, B., Hu, R., Brilakis, I., Bock, T., Pan, W., Carbonari, A., Castro, D., Mesa, H., Feng, C., Fischer, M., Brosque, C., Gonzalez, V., Hall, D., Ng, M. S., Kamat, V., Liang, C.-J., Lafhaj, Z., Pan, W., Pan, M., and Zhu, Z., editors, *Proceedings of the 39th International Symposium on Automation and Robotics in Construction*, pages 375–382, Bogot·, Colombia. International Association for Automation and Robotics in Construction (IAARC).

Yilmaz, N. and Ergen, E. (2024). Enhancing rfi analysis in construction projects: A comparative study of text clustering methods and visualization techniques. In *Proceedings of the 2024 European Conference on Computing in Construction*, volume 5 of *Computing in Construction*, Chania, Greece. European Council on Computing in

Construction.

Zhang, H., Lei, Y., Gui, L., Yang, M., He, Y., Wang, H., and Xu, R. (2024). CPPO: Continual learning for reinforcement learning with human feedback. In *The Twelfth International Conference on Learning Representations*.

Zhang, T., Bhatia, A., Pandya, D., Sahinidis, N. V., Cao, Y., and Flores-Cerrillo, J. (2020). Industrial text analytics for reliability with derivative-free optimization. *Computers & Chemical Engineering*, 135:106763.

Zhou, X., Yuan, Y., Yang, S., and Hao, J. (2025). Mentor: Guiding hierarchical reinforcement learning with human feedback and dynamic distance constraint. *IEEE Transactions on Emerging Topics in Computational Intelligence*.

Zuo, S., Wang, Z., Zhu, X., and Ou, Y. (2017). Continuous reinforcement learning from human demonstrations with integrated experience replay for autonomous driving. In *2017 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 2450–2455. IEEE.

Özoğul, C. and Ergen, E. (2024). Metadata extraction of rfis using natural language processing and machine learning algorithms. In *EC3 Conference 2024*.