Joint CSCE Construction Specialty & CRC Conference 2025
Conférence conjointe spécialisée en construction de la SCGC et CRC-2025

Montreal, Quebec
July 28-31, 2025 / *28-31 juillet 2025*

EVENT GRAPH OPTIMIZATION FOR REQUEST FOR INFORMATION (RFI) WITH HIERARCHICAL REINFORCEMENT LEARNING AND HUMAN FEEDBACK

S. Ahmadinasab[1], Y. Zhao[1], Y. Li[1], S. Sahyoun[2], M. Nik-Bakht[2], H. Ge[2], Y. Zeng[1]
and J. Yan[1]
[1]Concordia Institute for Information System Engineering
[2]Department of Building, Civil and Environmental Engineering
Concordia University, Montreal, Canada

ABSTRACT: Requests for Information (RFIs) are vital tools for stakeholder communication in construction project management, which aims to seek clarification, resolve ambiguities, and enhance project execution. Identifying the root causes of anomalies from RFIs is crucial for reducing costs and minimizing time delays. However, traditional Root Cause Analysis (RCA) methods often face challenges due to the complex and semi-structured nature of RFIs, resulting in limited performance that leaves room for improvement. In this study, we propose a novel approach utilizing human feedback-augmented reinforcement learning, which integrates expert insights as a core component to refine and optimize causal and temporal graphs generated by existing RCA methods. 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, by applying HRL and embedding HF into the RL loop, our method significantly improves the accuracy and reliability of causal and temporal graphs by effectively identifying and correcting errors within the graphs themselves.

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

1. INTRODUCTION

Construction projects rely heavily on documentation—such as contracts, drawings, and specifications—to guide their execution. However, these documents often fail to incorporate all factors critical for ensuring the project's constructability, leading to gaps, conflicts, or ambiguities that emerge during the construction phase. When these uncertainties arise, additional input is typically required from the design team to provide clarification (Song et al., 2009).

Despite advancements such as Building Information Modeling (BIM) (Jung and Joo, 2011) and design improvement tools (O'Connor and Koo, 2021), unforeseen challenges persist due to human error, project complexity, and the inherent uncertainties of construction work. To manage these gaps, RFIs have become essential tools in the construction industry, proving vital for resolving issues not identified during the design review process (Aibinu et al., 2020). An RFI is defined as "a formal written procedure initiated by the contractor seeking additional information or clarification for issues related to design, construction, and other contract documents" (Hanna et al., 2012). Common reasons for issuing an RFI include unclear specifications, plan contradictions, ambiguous construction documents, or unforeseen field conditions.

RFIs are semi-structured documents with predefined fields (e.g., project ID, date, query category) and unstructured inquiry text. The process starts with a team member submitting a query, which is reviewed by project managers or engineers who consult documents to provide a response, resolve the issue, and ensure project continuity. Extracting insights from RFIs can reveal early indicators of potential issues, project uncertainties, and risks (Lee and Yi, 2017). However, approximately 80% of construction-related information is embedded within documents such as RFIs, and professionals must manually search and interpret this information—a time-consuming and error-prone process (Abbaszadegan and Grau, 2015; Abdirad, 2022). Traditional manual methods, while useful, are inefficient for handling the high volume of RFIs found in large-scale projects. To streamline the extraction of useful information from RFIs, advanced Natural Language Processing (NLP) and text mining techniques—such as semantic analysis and clustering—have been adopted (Afzal et al., 2023; Jallan et al., 2019; Gheeseewan and Pudaruth, 2020). While these techniques help organize RFI data, they often cannot capture deeper causal and temporal relationships essential for understanding underlying issues and assessing project-wide impacts.

To understand the complex, interconnected nature of issues documented within RFIs, a structured approach to causal and temporal analysis is required. By generating causal and temporal graphs that depict underlying causal and temporal relationships among RFI issues, it is possible to identify recurring problems and trace them back to their origins. For example, a causal and temporal graph can link factors such as incomplete specifications or regulatory constraints to specific project events such as delayed inspections or material shortages. This approach provides a visual map of interconnected issues and supports informed decision-making based on the root causes of nonconformance. In construction, resolving RFIs quickly and accurately is essential for maintaining project timelines and budgets, making causal and temporal graph analysis a valuable tool for addressing systemic inefficiencies (Kelly and Ilozor, 2022). Previous studies emphasized the importance of causal and temporal graphs in addressing issues related to unclear or incomplete RFIs, which can lead to costly delays and project mismanagement (Philips-Ryder et al., 2013).

RFIs comprise elements such as events and factors that serve as foundational units for causal and temporal analysis. Events refer to actions or outcomes documented in the RFIs, such as design changes or inspection requests, while factors are underlying conditions—such as regulatory requirements or environmental constraints—that contribute to these events. Representing these components in a causal and temporal graph, with events and factors as nodes and causal or temporal links as edges, allows project teams to trace specific causes leading to particular effects visually. This structured representation enhances both problem-solving and decision-making.

Recent studies have emphasized using causal diagrams in construction for analyzing complex interdependencies and improving decision-making (Naser, 2023). Integrating causal analysis into simulation design enhances predictive modeling by preventing cognitive biases and ensuring robust parameter validation (Chen et al., 2023). However, current RFI analysis methods lack the sophistication to fully capture layered causal and temporal dependencies.

This study introduces an innovative approach that integrates HRL with HF to iteratively refine causal and temporal graphs for RFIs. While HRL with HF is well established in machine learning, its application to RFIs in the construction industry is novel, addressing the complexities of large-scale projects. Given the tendency of advanced language models to generate inaccurate outputs, human oversight remains crucial in refining causal representations (Bai et al., 2024). Our research contextualizes HRL for the construction industry, leveraging domain expertise to enhance the accuracy and relevance of causal and temporal insights. The contributions of this study are as follows:

- **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 RCA.

- **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 such as 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.

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. This HRL-based iterative framework provides a robust solution to refine and optimize causal and temporal graphs generated by existing RCA methods. The remainder of the paper is structured as follows: Section 2 reviews related work on causal and temporal relationship identification, RFIs, and RL. Section 3 details the methodology, including our HRL-based approach. Section 4 presents the experimental setup and results, followed by a conclusion and potential directions for future research in Section 5.

2. RELATED WORKS

Understanding causality and temporal relationships between events is critical to analyzing and improving processes in construction projects to address project delays, cost overruns, and quality issues. RFIs play a key role in documenting project ambiguities and facilitating communication, making them a valuable source for generating causal and temporal graphs. Foundational studies laid the groundwork for analyzing RFIs by identifying root causes and linking them to project outcomes, such as delays and rework costs(Hanna et al., 2012; Love et al., 2014). However, these methodologies heavily relied on expert input or manual verification, making them time-intensive and less scalable for large datasets.

Recent advancements in causal and temporal graph generation have leveraged NLP and event causality extraction to address the limitations of traditional approaches. Latent Dirichlet Allocation (LDA) was employed (Afzal et al., 2023) to extract patterns in RFI data, though their method lacked comprehensive comparison and assessment of clustering techniques. Studies such as (Gopalakrishnan et al., 2023) synthesized causal graphs from unstructured business texts but did not address domain-specific nuances, such as those in construction RFIs, which require iterative refinement. Recursive Neural Tensor Networks are introduced (Fischbach et al., 2021) for causality extraction, effective for detailed causality but limited by the complexity of parsing multi-causal events common in RFIs. Similarly, frameworks such as EventKGE (Li et al., 2023) and CausalKG (Jaimini and Sheth, 2022) use embedding and counterfactual reasoning to enhance causal graph transparency but lack adaptability for the sequential, feedback-driven refinement.

Recent advancements have also led to new semi-automatic systems for constructing causal graphs by synthesizing information from existing literature and expert knowledge(Cheng et al., 2020). The co-development of causal graphs through collaboration between domain experts and statistical modelers was introduced by (Kling, 2024), utilizing weighted false discovery rate (FDR)-adjusted p-values to guide the construction process via an iterative dialogue that refines the causal graph to align with both theoretical assumptions and empirical data.

A survey on event relation extraction identified challenges in handling multi-event dependencies(Liu et al., 2020). Similarly, HRL with HF has been applied for RCA in microservices (Wang et al., 2023), but it focuses on static service dependencies in controlled environments. Unlike Wang's model, our HRLHF framework is tailored for the dynamic, semi-structured nature of construction RFIs, incorporating iterative feedback to capture evolving project-specific complexities. However, many text analysis methods do not delve into complex, multi-layered causality critical, especially in construction RFIs. By incorporating HF and iterative refinement, the model ensures that causal and temporal graphs evolve dynamically, capturing project-specific complexities and dependencies while maintaining scalability for large datasets.

3. METHODOLOGY

Given a dataset of annotated text instances containing textual event descriptions and 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 an HRL framework integrated with an 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 Figure1, with interconnected components of the HRL framework and the iterative graph optimization process.

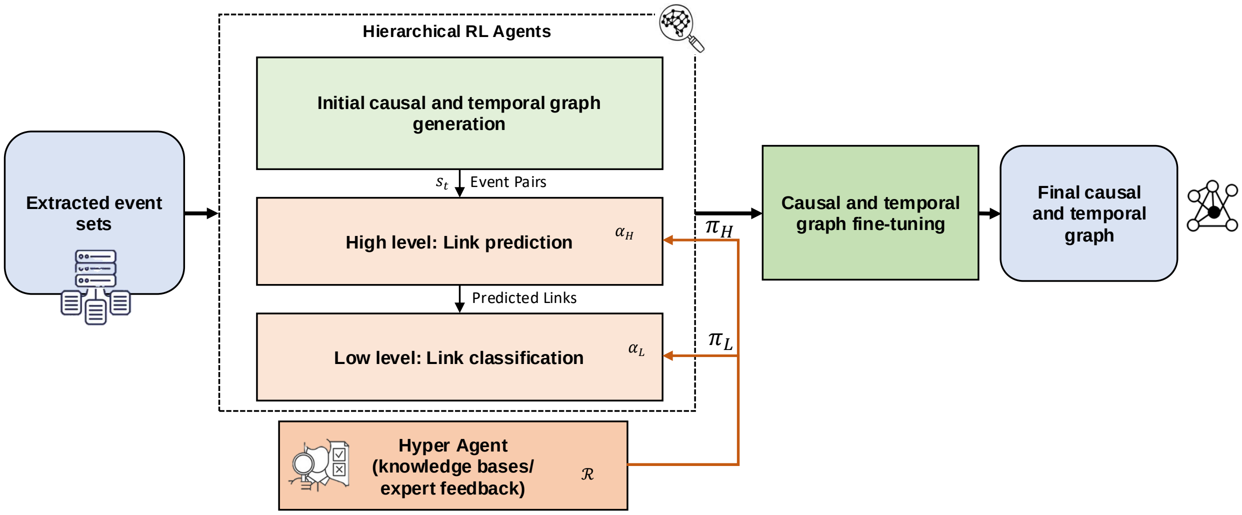
Hierarchical RL Agents
Initial causal and temporal graph generation
s_t
Event Pairs
High level: Link prediction
α_H
Predicted Links
Low level: Link classification
α_L
Extracted event sets
π_H
π_L
Causal and temporal graph fine-tuning
Final causal and temporal graph
Hyper Agent (knowledge bases/ expert feedback)
$\mathcal{R}$

Figure 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 HRL approach employs a hierarchical approach to improve causal and temporal graph generation: **High-Level Agents:** These agents predict links between extracted event sets, determining whether and where connections should be established or removed within the graph. This decision-making process is guided by a Markov Decision Process (MDP), where states represent the current configuration of causal and temporal links, and actions (link/unlink) determine structural adjustments.

The MDP is defined as a tuple (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.

The reward function R(s, a) assigns rewards based on how well the generated causal and temporal structure aligns with expert insights, penalizing incorrect link types and overly dense graphs. γ is the discount factor, controlling the weight of future rewards. The high-level policy's Q-values, $Q(s,a)$, are updated using Q-learning (Jang et al., 2019; Sutton and Barto, 2018), where each action a taken in the state s is rewarded based on improvements in graph accuracy, with the reward calculated as:

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

where $Q(s,a)$ is the current estimate of the value of taking action in the state, $\max_{a'} Q(s',a')$ is the maximum Q-value for the next state, α is the learning rate, r the reward for taking action, and γ the discount factor.

Low-Level Agents: Conditional on decisions made by high-level agents, low-level agents classify link types (CLINK, TLINK) and refine connection details, optimizing causal and temporal relationships while ensuring accuracy and domain compliance.

3.2 Reward Model for Graph Refinement

The reward model assesses the graph quality by comparing it to the ground truth, which consists of expert-annotated causal and temporal relationships derived from the dataset. It assigns rewards for correctly classified CLINKs and TLINKs, penalties for false positives (adding wrong edges) and false negatives (missing true edges), maintenance rewards (R_{M}) for retaining accurate edges, and density penalties (P_{D}) to discourage over-linking.

The total reward is computed as:

[2] $R = w(N_{\text{CL}}) + w(N_{\text{TL}}) + w(N_{\text{FP}}) + w(N_{\text{FN}}) + R_{\text{M}} + P_{\text{D}},$

where each component is defined as:

[3] $R_{\text{M}} = (N_{\text{CL}} + N_{\text{TL}}) \times w_{\text{M}},$

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

where w is the weight parameter for each reward/penalty component, w_{D} is the penalty for excessive graph density, ρ is the graph density, and ρ_{t} is the threshold for graph density.

This formula penalizes excessive graph density, preventing the model from unnecessarily over-linking nodes. The coefficient of 5 was selected to adjust the penalty based on how much the graph exceeds the predefined density threshold. This ensures an optimal structure by minimizing extraneous edges that could introduce incorrect causal or temporal relationships.

The weight parameters for the reward function were determined through empirical tuning to achieve an optimal balance between different objectives. Table 1 lists the reward and penalty weights used in the experiments.

Table 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 experiments, ensuring the model learns efficiently while avoiding unnecessary link creation. The balance between rewards and penalties ensures that the RL agents learn to refine the graph optimally, enhancing accuracy while minimizing errors.

3.3 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 training, 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 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 ensure that HF is leveraged to enhance causal and temporal inference.

4. EXPERIMENTS AND RESULTS

4.1 Experimental Setup

Datasets We evaluate our HRLHF model on the widely adopted Causal-TimeBank dataset (Mirza et al., 2014), which comprises 184 documents and 6813 events. Among these, 318 event pairs are annotated with causal relations, while 5,118 event pairs are annotated with temporal relations. While the Causal-TimeBank dataset is not specific to the construction domain, it provides a structured benchmark for evaluating causal-temporal inference that is well-suited for evaluating causal and temporal inference models. In this study, we deliberately scope our contribution to the task of event graph optimization based on structured textual data, which represents one of the core challenges in Request for Information (RFI) analysis. Our proposed HRLHF framework is designed to operate on such structured event data, assuming that event extraction and initial graph construction have been completed as part of the preprocessing pipeline. Since this work focuses on optimizing event graphs derived from structured textual data, it does not extend to interpreting or extracting information from construction drawings, images, or other non-textual formats. Accordingly, using the Causal-TimeBank dataset allows us to isolate and validate the performance of the graph optimization mechanism. Following the standard procedure in causal and temporal relation extraction tasks, we treat the annotated causal and temporal links as the target relationships.

Experimental Design and Hyperparameters To optimize performance, the model underwent extensive fine-tuning across 72 experiments, testing different combinations of key hyperparameters: learning rate (`lr`), discount factor (`df`), exploration rate (`exr`), and exploration decay (`exd`). The final configuration, 0.5, 0.95, 0.99, and 0.98, accordingly, was selected for its balance between precision, recall, and F1 Score. The model has trained over 200, 500, and 1000 iterations, gradually reducing exploration to support effective convergence, particularly under varying graph error conditions.

Evaluation Metrics To evaluate the quality of the refined graph G' relative to the true graph G_{true}, we use three key metrics: Precision, Recall, and F1 Score by comparing the results with the initial graph G_{init} as follows:

[5] Precision $= \frac{\text{TP}}{\text{TP}+\text{FP}}$,

where TP is True Positive (edges in both G_{final} and G_{true}) and FP is False Positives (edges in G_{final} but not G_{true}).

[6] Recall $= \frac{\text{TP}}{\text{TP}+\text{FN}}$,

where FN is False Negative (edges in G_{true} but missing from G_{final}).

[7] $\text{F1 Score} = 2 \cdot \frac{\text{Precision}\cdot\text{Recall}}{\text{Precision}+\text{Recall}}$

4.2 Results

This section evaluates the performance of our HRLHF model on the Causal-TimeBank dataset. We also visualize cumulative rewards and reward convergence throughout the training process to analyze the model's learning behavior. The objective of the experiment is to assess the model's capability to refine an initial error-induced graph into a structure that closely aligns with the true graph. So, we compared the initial graph, which contains synthetic errors, with the final graph produced after applying the HRL framework.

The evaluation results are presented in Tables 2 and 3, which compare the performance of the initial graph with some errors before HRLHF optimization (Table 2) to that of the final graphs after HRLHF (Table 3). The average performance was obtained across 30 experiments to ensure reliability and generalizability. Each experiment was conducted on graphs with varying error levels, and the metrics were averaged to assess the framework's effectiveness. The results demonstrate that HRLHF significantly improves graph accuracy under various error conditions.

To simulate realistic errors, two types of errors were introduced:

- Misprediction errors refer to the random addition or removal of edges between nodes in the graph.
- Misclassification errors refer to modifying the type of existing edges which could lead to incorrect causal or temporal relationships.

Table 2: 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 3: 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

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

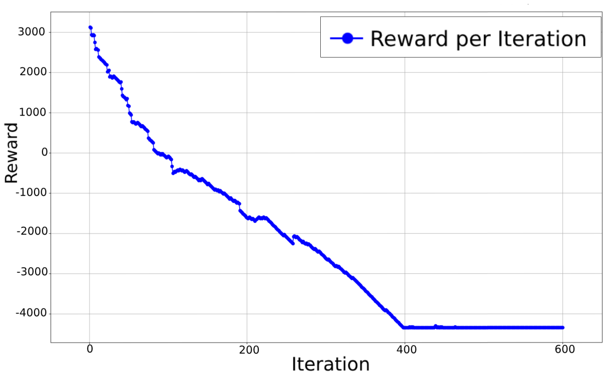

Figure 2: Immediate reward through the training process, which converges after 400 iterations.

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% 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.

Table 4 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: 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 2 illustrates a gradual decline in reward values as the model stabilizes and approaches optimal solutions. Initially, a sharp decrease in rewards indicates that the model is exploring various strategies and learning from less optimal decisions. As the iterations progress beyond 200, the reward decrease becomes more gradual, indicating a refining of choices and adaptation towards more effective strategies. This trend continues until about the 600th iteration, where the rewards level off, maintaining a near-constant value, which suggests that the model has potentially reached a stable policy. The flat line in the later sections of the graph signifies minimal changes in reward, indicating convergence to an optimal or near-optimal policy, as the immediate rewards reduce to less than a small threshold for over 400 iterations, signifying a significant stabilization in learned behaviors.

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.

The F1 Score and precision improvements result from HF, which helps the model refine predictions iteratively. Figure 3 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. 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.

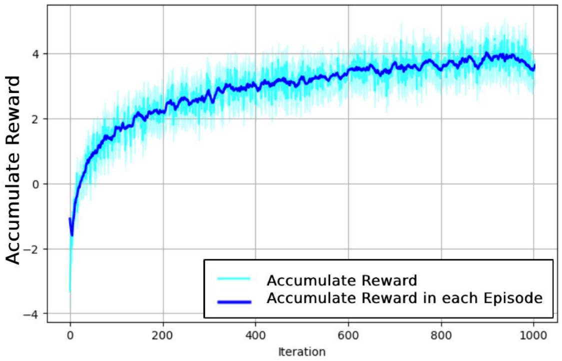

Figure 3: Accumulated rewards over 1000 iterations to show the RL performance progression.

5. CONCLUSION AND FUTURE WORKS

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. We evaluated the individual contributions of HRL and HF to refining causal and temporal graphs. When augmented with HF, HRL significantly improves graph structure by correcting missing and inaccurate edges through direct human intervention. Precision, recall, and F1 Score metrics demonstrate the effectiveness of our approach in optimizing causal and temporal graphs. By integrating domain expertise into automated learning, our HRL-HF framework enhances graph accuracy and interpretability, making it applicable to construction data analysis and process mining. Future work will focus on scaling the model to domain-specific datasets, assessing its performance on complex, diverse data, and extending its applicability to healthcare, financial systems, and natural disaster analysis.

REFERENCES

Abbaszadegan, A. and Grau, D. (2015). Assessing the influence of automated data analytics on cost and schedule performance. *Procedia Engineering*, 123:3–6.

Abdirad, H. (2022). Managing digital integration routines in engineering firms: Cases of disruptive bim cloud collaboration protocols. *Journal of Management in Engineering*, 38(1):05021012.

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.

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.

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

Cheng, K., Cui, X., Guo, S., Yang, J., and Wu, X. (2020). Semi-automatic causal graph construction system. *Proceedings of the 3rd International Conference on Computer Science and Software Engineering*.

Fischbach, J., Springer, T., Frattini, J., Femmer, H., Vogelsang, A., and Mendez, D. (2021). Fine-grained causality extraction from natural language requirements using recursive neural tensor networks. In *2021 IEEE 29th International Requirements Engineering Conference Workshops (REW)*, pages 60–69. IEEE.

Gheeseewan, H. and Pudaruth, S. (2020). Categorisation of computer science research papers using supervised machine learning techniques. *Int. J. Com. Dig. Sys*, 9(6).

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.

Jaimini, U. and Sheth, A. (2022). Causalkg: Causal knowledge graph explainability using interventional and counterfactual reasoning. *IEEE Internet Computing*, 26(1):43–50.

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.

Jang, B., Kim, M., Harerimana, G., and Kim, J. W. (2019). Q-learning algorithms: A comprehensive classification and applications. *IEEE Access*, 7:133653–133667.

Jung, Y. and Joo, M. (2011). Building information modelling (bim) framework for practical implementation. *Automation in Construction*, 20:126–133.

Kelly, D. and Ilozor, B. (2022). Performance outcome assessment of the integrated project delivery (ipd) method for commercial construction projects in usa. *International Journal of Construction Management*, 22(14):2808–2816.

Kling, E. Y. (2024). Co-developing causal graphs with domain experts guided by weighted fdr-adjusted p-values.

Lee, J. and Yi, J.-S. (2017). Predicting project's uncertainty risk in the bidding process by integrating unstructured text data and structured numerical data using text mining. *Applied Sciences*, 7(11):1141.

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.

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.

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., 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.

Naser, M. (2023). Application of causal diagrams in civil engineering. *ArXiv*, abs/2306.15834.

O'Connor, J. T. and Koo, H. J. (2021). Proactive design quality assessment tool for building projects. *Journal of Construction Engineering and Management*, 147(2):04020174.

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.

Song, L., Mohamed, Y., and AbouRizk, S. M. (2009). Early contractor involvement in design and its impact on construction schedule performance. *Journal of management in engineering*, 25(1):12–20.

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

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.