AUTOMATED SCHEDULE ENRICHMENT FROM DAILY PROGRESS REPORTS: A BI-DIRECTIONAL NEURO-SYMBOLIC AI APPROACH

Sudheer Kumar Nanduri, Venkata Santosh Kumar Delhi
Indian Institute of Technology Bombay, India

Abstract

Construction progress monitoring requires integrating daily progress reports (DPRs) with baseline schedules for productivity analysis and risk management. Manual integration is unscalable due to heterogeneous formats and inconsistent terminologies. This paper presents a neuro-symbolic system that enriches schedules from the DPR files utilizing the large language models and knowledge graphs. The system learns construction terminology from schedule and DPRs, adapts confidence weighting, and validates outputs against project constraints. Tested on a real construction project with 173 DPRs and 1000 validation activities, it achieved a median confidence of 0.825 and zero constraint violations.

Introduction

Construction schedules are central to project planning, monitoring, and control (Project Management Institute, 2021), yet they are rarely updated using the execution data collected during the construction (Olawale & Sun, 2015). Intermittent and partial schedules are used in the routine project communications, losing the golden thread of workflow information across multiple digital documents (Adekunle et al., 2022). Daily progress reports (DPRs) capture detailed fieldwork plans, completed and delayed activities, and exact quantities, resources, and locations. But they typically remain disconnected from the baseline schedules.

Automating the integration of DPRs with schedules, referred to as schedule enrichment in this work, remains a challenge due to heterogeneity arising from document origins. DPRs are semi-structured documents, and the terminology used varies significantly across projects, contractors, and engineers. In contrast, schedules are more formal, structured, and governed by strict temporal and precedence constraints. Field implementations may sometimes relax the strictness, but the precedence information remains logical. Bridging these two information sources requires both semantic flexibility and logical consistency.

Existing automated approaches address part of the problem. Language models can recognize semantic similarity between activity descriptions, but cannot reliably enforce temporal or causal constraints. Rule-based systems provide logical rigor but fail when faced with vocabulary variations and incomplete ontologies. Sequential combinations of these approaches remain brittle, as validation occurs after matching decisions are made.

This paper proposes a neuro-symbolic approach that integrates neural semantic reasoning with symbolic constraint validation during inference. The system learns construction vocabulary automatically from successful alignments and adaptively balances neural and symbolic contributions as knowledge accumulates. The approach is validated on real project data and demonstrates that bi-directional neuro-symbolic integration enables scalable, interpretable, and constraint-aware schedule enrichment.

Related Work

Project monitoring and automated schedule updating are actively researched topics in the construction automation domain (Turkan et al., 2012). Existing approaches rely heavily on building information modeling (BIM) with visual data analytics (Golparvar-Fard et al., 2015) or point clouds (Park & Cho, 2022). While model-based approaches are effective in geometric progress tracking, they are less suited for capturing workflow semantics and also face adoption challenges (Saka & Chan, 2020).

These limitations motivate a shift towards knowledge-based approaches that operate directly on documents. Natural language processing (NLP)- based approaches were explored for extracting activity information from schedules (Zhao et al., 2020) and for bridging schedule activities to the Uniformat (Jung et al., 2024) standard, thereby standardizing the ontology.

Transformer-based approaches were found to be valuable for aligning schedules with field reports and for generating look-ahead planning (Amer et al., 2021, 2023). Large Language Models (LLMs) have demonstrated strong performance in understanding domain-specific vocabulary with minimal task-specific engineering. From generating schedules autonomously (Al-Sinan et al., 2024) to identifying constraints from field meeting transcripts (He et al., 2024). Although LLMs have shown generative capabilities in semantics, the ontology is the primary driver of the logic and validation, which LLMs do not explicitly model.

In contrast, graph-based methods (C. Wu et al., 2023) provide a structured mechanism for enriching existing knowledge, either from schedules or from constraints. Domain-specific knowledge graphs are graph-based

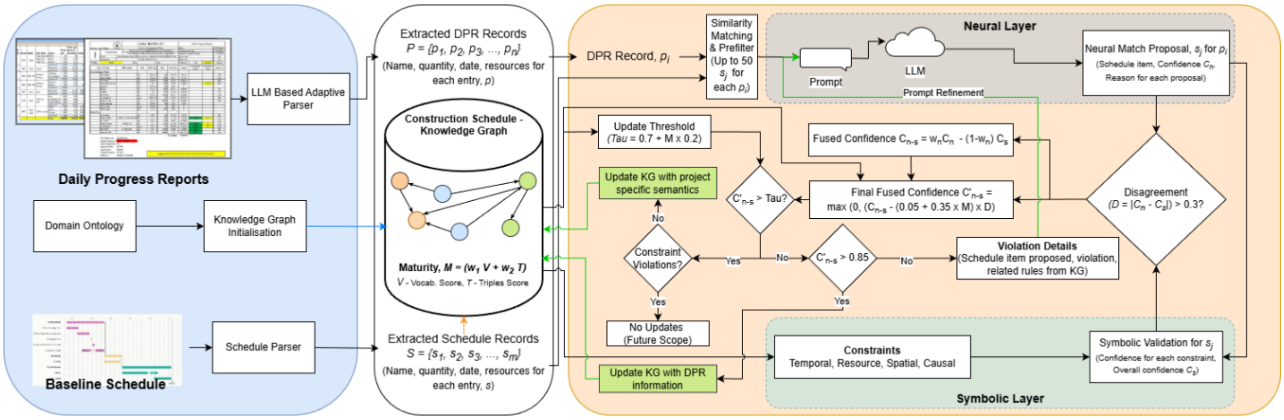

Figure 1 Overall System Architecture

approaches that simultaneously store the project information and allow for constraint application mapping very easily with a construction schedule (S. Wu et al., 2024).

Further, LLMs are shown to work well with knowledge graphs (KG) (Cai et al., 2025), even within the domain of construction automation (Li et al., 2024). However, most current efforts remain unidirectional flows that allow a neural system to extract data for a symbolic system to process, limiting the system's ability to incrementally gather knowledge from daily changes in the field.

Neuro-symbolic AI offers a more robust approach. Recent work in neuro-symbolic AI has proposed multiple integration strategies ranging from loose coupling to tightly embedded symbolic reasoning (Kautz, 2022). The current methods of NLP lack the logical rigor needed for understanding schedules and field documentation without engineering. Language model approaches demonstrate strong semantic understanding of unstructured information but lack mechanisms to enforce constraints required for schedule correctness. Rule-based and ontology-driven approaches enable constraint checking and logical consistency but depend on manually curated vocabularies and struggle with heterogeneous, informal field terminologies.

Thus, the specific research gap lies in the absence of a Type 6 (Neuro[Symbolic]) framework that embeds symbolic reasoning as a subroutine within a neural process loop. Existing research does not address automated vocabulary learning or adaptive integration, in which the system transitions the governance as domain knowledge evolves. As a result, there is currently no method that enables scalable, constraint-aware alignment of DPRs with schedules while continuously learning project-specific terminology.

Methodology

To address the identified research gap, this paper proposes a neuro-symbolic methodology for automated construction schedule enrichment from daily progress reports. The method integrates neural semantic reasoning with symbolic constraint validation in a unified inference process, enabling robust alignment between DPR data and schedule data.

System Architecture

The proposed methodology, structured and presented in Figure 1, is designed as a Bi-directional Neuro[Symbolic] (Type 6) (Kautz, 2022) architecture. This approach balances the semantic capabilities of LLMs and the validation capabilities of KGs in an iterative loop. The overall architecture has three interacting layers:

1. Data preparation layer responsible for parsing DPR files, scheduling data, and initializing the knowledge graph.
2. The Knowledge graph layer, which gets enriched through the processes, can be used to enrich the schedule after processing DPR.
3. Neuro-Symbolic matching layer responsible for candidate DPR-schedule alignment proposal, validation, and refinement.

Problem formulation

Let

$P = \{p_1, p_2, p_3, \ldots, p_n\}$ denote the set of extracted DPR records,

$S = \{s, s_2, s_3, \ldots, s_m\}$ denote the set of baseline schedule tasks, and

Each DPR activity p_i is associated with observed attributes, including execution date, quantity, location, and resource usage. Each scheduled task s_j is associated with planned attributes, including time windows, locations, quantities, and precedence relationships.

The objective is to determine a mapping function.

$$F : P \rightarrow S$$

that aligns the DPR activities with corresponding schedule tasks while maintaining logical consistency and enabling schedule enrichment. Constraint information used during alignment is derived from schedule-defined constraints and from observed constraints inferred from DPR execution data. The system validates proposed vocabulary and alignments against available constraints and incrementally augments the knowledge graph with newly observed data when sufficient evidence is present. These learned mappings are subsequently used to refine future alignments and to enrich the baseline schedule without manual ontology engineering.

Data Preparation Layer

Daily Progress Report Parsing

In this work, we utilised DPR files generated at the project level by the contractor's planning office and submitted to the cost consultant. All reports originate from a single organizational source and use a common spreadsheet template, reducing the heterogeneity of file type choices and broader design choices. However, different engineers introduced varying column ordering, multiple data regions per sheet, mixed data types, merged cells and colours, comments along with the data, and hidden row data, apart from progress details. In case DPRs originate from multiple sources with different formats, the same parsing pipeline described below can be applied to each source.

We followed a semi-automated approach for parser scripting. The first 3-5 DPR files are sampled for inspection to read raw cell values. Known patterns of the document structure that can be ignored are coded to allow focusing on the header, data presence, and hierarchy markers from merged cells. The extracted details are used to construct a detailed prompt for Large Language Models (LLMs) to generate code for targeted data parsing.

The LLM-generated script is run on all the DPR files for data parsing and extraction. Each parsed record is checked for activity name, quantity, date, and location hierarchy, and missing data is used to calculate a quality score for the LLM-generated script. The quality score is used to generate new code versions for improved data extraction, with details of the previous run's errors and additional directions added to the prompt.

Schedule Loading and Knowledge Graph Initialisation

The baseline schedule is parsed for task definitions, structure, temporal constraints, quantities, and relationships. Simple natural language processing steps, such as case correction, punctuation removal, and lemmatization, are applied to the extracted textual data. The scheduled tasks and their relationships are represented as entities and relations within the knowledge graph.

The knowledge graph is initialised as a formal RDF graph. Base ontology is created with domain-specific semantics comprising 16 common activity types (excavation, PCC, etc.), 12 logic-based precedence constraints (reinforcement before shuttering, etc.), and 4 base synonyms (shuttering synonym of formwork, etc.), forming around 80 triples.

For each schedule task, the graph instantiates a unique entity, mapping it to its corresponding activity type and its temporal predecessors as graph edges. One project-specific problem was that the baseline schedule is activity-based but not delineated by location, whereas DPRs present activity and location hierarchy. At the initial schedule ingestion phase, enforcing location is not useful, and so the location constraint is kept for semantic hinting to LLMs and only soft matching in the KG.

Since we do not have the project-specific knowledge at the start, the knowledge graph is set up with a maturity score to guide the transition from purely neural to neuro-symbolic. This setup allowed us to achieve better semantic performance without encountering cold-start issues. The equations below represent the maturity formula built for the purpose.

$$Maturity\ Score, M = w_1 * V + w_2 * T \quad (1)$$

$$Vocabulary\ Score, V = min\left(1.0, \frac{v_l}{v_T}\right) \quad (2)$$

$$Triples\ Score, T = min\left(1.0, \frac{t_l}{t_T}\right) \quad (3)$$

Where,

v_l = Learned mappings beyond the baseline KG initialisation (0 mappings),

v_T = Threshold mappings (200 mappings) – considered to be fully mature,

t_l = Learned triples beyond the baseline KG initialization (86 triples),

t_T = Threshold triples (10,000 triples),

w_1, w_2 = Weightage for vocabulary score and triples score. (0.7, 0.3 respectively)

The weights are empirically determined, specific to the project, based on the available data. The thresholds are considered based on domain knowledge. Roughly 200 unique activities and 10,000 triples can be considered for the construction activities.

The KG is considered immature when its score is less than 0.1, and most subsequent layers tend to use the neural system in such cases. As the KG becomes more enriched, its maturity increases, and the dependency on the symbolic component also increases.

Neuro-Symbolic Layer

This process follows each DPR activity in a multi-step alignment process.

Neural Proposal Stage

The activity description of each task pi, a list of leaf tasks (non-summary tasks), and a prompt to select the best matching task based on semantic, spatial context, and hierarchy are sent to an LLM.

One specific detail of the task list is the workflow based on the KG's maturity. If the KG is immature, the top 50 schedule tasks are utilised. If the KG is mature, then a symbolic pre-filtering stage is applied. The filters include activity type match, keyword overlap, and trade separation, which are used to filter the tasks. The schedule data lacked separation of MEP concerns, whereas the DPR provides detailed separation. So, immature KG will not know the correct MEP trade for pre-filtering by activity type, and so this part is explicitly added. If filtering is successful, the top 50 tasks are considered for similarity matching.

The LLM returns a scheduled task that matches the prompt, along with its confidence and reasoning. The confidence value is the neural system confidence (C_{neural}), ranging from 0 to 1.

Symbolic Validation Stage

The proposed schedule element is validated against four constraints in the KG.

1. Temporal constraint: The DPR task must fall within a temporal window with ±7 days over the scheduled start and end dates of the task. Additional validation regarding the completion of the prerequisite tasks before the activity's date
2. Resource constraint: The DPR task achieved cumulative quantity must fall within a 30% range of the scheduled task
3. Spatial constraint: The DPR task overlaps with activity locations in the KG
4. Causal constraint: The DPR task preceding activities must be completed. Check if earlier matched DPR activity exists.

All violations are treated as soft violations, as weather issues, resource constraints, scope changes, and other factors can occur during field execution. We assigned empirical confidence penalties (0.9 when valid, 0.3-0.4 when invalid), calibrated on a held-out subset of 50 DPR records.

To assess sensitivity, the penalties were uniformly varied by 20%. The median fused confidence ranged from 0.61 (for -20%) to 0.78 (+20%). A span of 0.17 points. The high-confidence alignment rate varied from 1.5% to 11.0%. This moderate variation suggested that exact penalty calibration is not a critical operation, and the system remains stable under reasonable parameter changes. The mean aggregated confidence across all four constraint dimensions is considered the symbolic system confidence ($C_{symbolic}$), ranging between 0 and 1.

Disagreement Resolution

The absolute difference between the neural and symbolic system confidence is considered Disagreement (D) if it exceeds 0.3. If there is a disagreement, a fused confidence score is calculated for processing.

$$D = |C_{neural} - C_{symbolic}| \quad (4)$$

$$C_{n-s} = w_{neural} * C_{neural} + (1 - w_{neural}) * C_{symbolic} \quad (5)$$

$$w_{neural} = 0.85 - (\mathrm{M} * 0.35) \quad (6)$$

Where,

D is the disagreement between neural and symbolic components,

C_{n-s} is the Fused Confidence Score of the system,

w_{neural} is the weight for the neural component, derived from the Maturity Score, M, described in (1)

C_{neural} & $C_{symbolic}$ are the neural and symbolic confidence values

The KG maturity score is inversely correlated with the symbolic system confidence. Thus, when KG is immature, the neural system is preferred for fused confidence. If the KG is mature, and there is a disagreement between the two components, the fused score is calculated with a penalty term based on the KG maturity score.

In practical terms, an immature KG keeps the workflow neural-heavy, the candidate selection remains broad, and the fused confidence is driven more by the LLM. As the KG matures, the workflow shifts towards symbolic governance through pre-filtering and stronger constraint weighting, typically producing lower but more reliable confidence scores.

$$Penalty\_strength, p = 0.05 + (M * 0.15) \quad (7)$$

$$Penalty, P = Penalty\_strength, p * Disagreement, D \quad (8)$$

$$C'_{n-s} = \max(0.0, (C_{n-s} - P)) \quad (9)$$

Where,

p is the penalty strength calculated over the Maturity Score S described in (1),

P is the penalty value, and

C'_{n-s} is the Final Fused Confidence Score.

The final fused confidence score is the decisive variable for the iterative refinement step.

Iterative refinement

If the fused confidence exceeds 0.85, the DPR is aligned with the schedule activity and updates the KG.

A variable threshold, τ, as detailed in (10), is used to determine whether to map the vocabulary and enrich the KG.

$$\tau = 0.7 + M * 0.2 \quad (10)$$

Where,

τ is the variable threshold calculated from the Maturity Score M mentioned in (1).

The final fused confidence score (C'_{n-s}) is compared against the threshold τ. If the confidence exceeds the threshold and there are no constraint violations, the DPR element is utilised for vocabulary mapping and for the DPR entry as a record of the progress of the actual schedule. Even if the confidence is below the threshold but greater than 0.85, the DPR entry is still recorded as a progress entry, but no vocabulary mapping will be made. If there are violations, but the confidence score is high, the constraints can be updated. But this step will require a detailed study of which constraints can be safely updated and which should be left for future work.

Both the above steps of threshold verification and updating the KG form the core of forming a $Neural -> Symbolic$ learning system. For the DPR activity p_i and the proposed schedule activity s_i, the fused confidence scores, with breakdowns, constraint violations, and other relevant details, are collected and sent back to the LLM for the next alignment suggestion, forming a $Symbolic -> Neural$ constraint enforcement system. To prevent the system from entering an infinite loop, iterative refinement stops after 3 rounds.

Experimental design

System Design Validation

To validate the model, the pipeline is run on the schedule and applied to 1000 activities from the DPRs. Six validation tests were conducted on the research claims. In addition, 105 DPR-schedule alignments (from the ground truth dataset, stratified across all three buildings) were manually verified by a domain expert. Metrics like precision and F1 score were computed at four evaluation levels: Strict task-UID match, building-aware match for correct activity type and building, activity type-only match, and coverage, where any match is produced when one is expected.

1. Neuro-Symbolic Integration – Integration logs were used to count events by direction between neural and symbolic systems, and measured the refinement success rates by comparing confidence scores. Considered a pass if we observe > 100 events with > 70% refinement success.
2. Bidirectional Learning – Alignment confidence scores were split into early (first 50%) and late (last 50%) groups, and the mean confidence change was calculated; improvement at the late stage was considered a pass if it exceeded 1%. Also, the vocabulary counts are fit to linear regression to detect trends.
3. Knowledge Accumulation – KG statistics before and after pipeline execution were measured as the growth factor in triples relative to the initial stage. If the growth factor is > 10 and the vocabulary mappings increase by> 50, then it is considered a pass. If the growth factor increases by 50, it is regarded as strong evidence
4. Entity Resolution Effectiveness – Entity counts before and after deduplication are compared. If the ratio is more than 50%, it is considered a pass, and it is considered strong if the ratio > 70%.
5. Symbolic Inference Performance – Triples discovered by the reasoning system and logical conflicts found are measured. The validation passed if inferred triples are >1000, and conflicts < 5%.

Baseline and Ablation Studies

For baseline studies, we implemented the same three approaches from scratch and executed them on the sample.

1. Pure Neural (LLM) – Direct LLM matching without KG validation or iterative refinement. We performed 10 LLM calls, each aligning one DPR activity to the full schedule. A smaller random sample of DPR activities is chosen to isolate and assess the LLM behaviour for relative trends rather than a full-scale evaluation.
2. Pure Symbolic (Rules) – String matching (and Jaccard similarity of keyword overlap executed on all 100 activity samples.
3. Traditional NLP (TF-IDF) –Vectorization and cosine similarity matching-based approach on all 1000 activities.

For ablations, we adopted an analytical approach. Each component's contribution is used to compute the degraded confidence if it were absent.

Learning Trajectory Verification

To verify the learning trajectory, we rerun the pipeline on the dataset 10 times, with a sample size of 10 in each run. In each iteration, the activity batch is randomized. The KG maturity, average confidence, and vocabulary learned are extracted at each iteration, and the final KG serves as the starting point for the next. We computed the maturity growth between the final and first iteration, and the confidence improvement ratio. Successful learning must show at least a 1% increase in confidence and a monotonically increasing level of maturity.

Results

System Design Validation

Table 1. System Design Validation Results

Test	Evidence	Metrics
Neuro-Symbolic Integration	Moderate	775 integration events, 100% refinement
Bi-directional Learning	Weak (single pass)	-7.7% confidence change (0.838 to 0.774); 0 vocabulary at pre-populated KG; 375 vocabulary mappings acquired during processing
Knowledge Accumulation	Strong	KG grew from 86 ontology triples to 39,620 (456x); alignment-driven growth: +4660 triples from 34,600 post-schedule baselines
Entity resolution	Strong	1139 activity variants reduced to 134 canonical entities (88% compression); 760 resolution events across exact (58%), LLM (34%), and similarity (3%) methods
Symbolic inference	Strong	28,975 triples inferred; 0 logical conflicts detected

On the manually verified subset, the precision and F1 scores for activity-type match are 0.930 and 0.815; for building-aware match, 0.338 and 0.400; for Strict task-UID matches, 0.141 and 0.189; and for coverage matches, 1.000 and 0.850. Across all 1000 alignments, 769 were matched, yielding a 76.9% match rate. Most strict-UID errors were not semantic failures. 56 of 61 wrong matches still identified the correct activity type, with errors dominated by cross-building confusions (42) and granularity mismatches (14). These results indicate strong semantic recognition, with the remaining difficulty concentrated in spatial disambiguation.

The bidirectional learning row in Table 1 reflects an early vocabulary saturation. The confidence decrease reflects the KG maturity (Eq. 6), penalizing matches that lack full constraint support. Thus, the negative confidence change reflects a calibration effect rather than evidence against bidirectional learning. The broader evidence for bidirectional learning comes from the same run's 375 vocabulary mappings acquired during DPR processing.

Details of the symbolic inference validation suggest that the symbolic reasoning primarily generated triples of transitive closure over the precedence relationship (about 73.8% of the total KG content), suggesting that implicit knowledge discovery amplifies explicit data. However, the absence of conflicts indicates the need to reexamine the precedence graph derived from the validated schedule. Further verification is needed using synthetic, contradictory data.

Baseline & Ablation Comparison

The baseline comparison across approaches is presented in Figure 2. The median confidence is higher than the mean, indicating a bimodal distribution with many high-quality matches. The pure LLM approach performs comparably but has lower mean and median values than the proposed approach. Higher confidence (10% compared to 4.3% of ours) in the LLM approach might suggest the lack of constraint validation, which is derived from the symbolic component in our system. The purely symbolic approach, understandably, fails to handle vocabulary variation. Still, the traditional NLP implementation treats pairs as minimal datasets to measure similarity and cannot be applied directly. However, this doesn't affect the overall implementation and results.

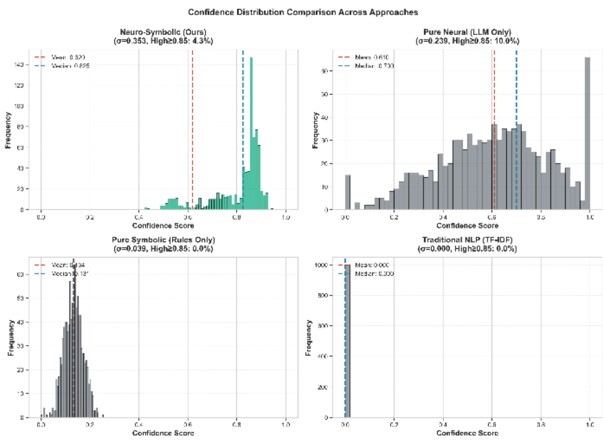

Figure 2 Baseline Comparison Results

The ablation study results are presented in Figure 3. The top contributors are KG validation, vocabulary learning, and iterative refinement, which together account for nearly 70% of the system's performance. All three are mechanisms of the neuro-symbolic system, confirming the system's synergy. Of the 769 alignments requiring refinement, KG validation successfully identified inconsistencies and is the most valuable component.

Qualitative Analysis

Representative examples from the evaluation illustrate the system's practical behaviour. A DPR entry reporting "Waterproofing" activity for Building C (Terrace Floor, Super-Structure section) was matched to the schedule task "Terrace Floor water proofing", with a fused confidence of 0.949 (neural 0.95, symbolic 0.947), indicating semantic agreement, temporal validity, spatial consistency, and precedence satisfaction. In a contrasting case, the DPR entry "Sunk Area Filling" (for building B) was matched to "PCC" with a fused confidence of 0.758 (neural 0.90, symbolic 0.0), resulting in a disagreement and flagging for expert review. 188 such cases were identified of the 1000 processed alignments, demonstrating the system's ability to surface uncertain matches rather than silently accepting them. The KG also learned 375 vocabulary mappings, including synonyms such as shuttering for formwork, typographical errors like conctere for concrete, and field-specific terms like UGWT Raft, mapped to the canonical Underground Water Tank activity type, which persisted in the KG without requiring further LLM calls.

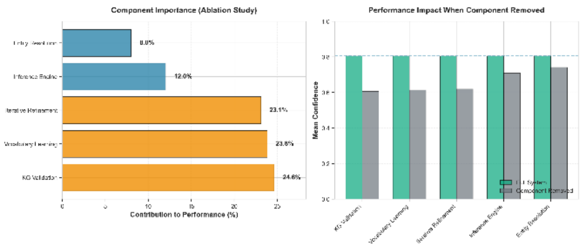

Figure 3 Ablation Study Results

Learning Trajectory Verification

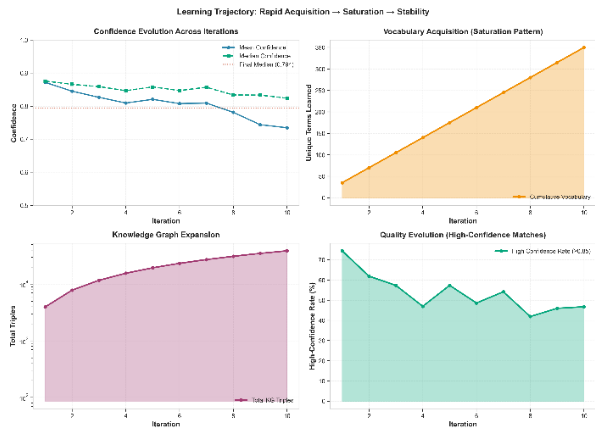

Figure 4 Learning Trajectory Verification

The learning trajectory results over 10 iterations are presented in Figure 4. Overall, the KG maturity shows a +68% increase, with 7 terms and 44 KG triples learned. The confidence decreases by 1.7%, with intermittent dips, possibly due to adaptation to edge cases. The initial iterations also show high values, suggesting that the initial sending with the schedule and the first iteration itself rapidly accumulates valuable project-specific knowledge.

Figure 5 shows that most KG growth comes from schedule-derived structural scaffolding, while DPR alignment provides a smaller but semantically important set of learned mappings and observations.

Discussion

The results demonstrate that the proposed neuro-symbolic system can appropriately align daily progress reports with baseline schedules and regulate confidence in logically

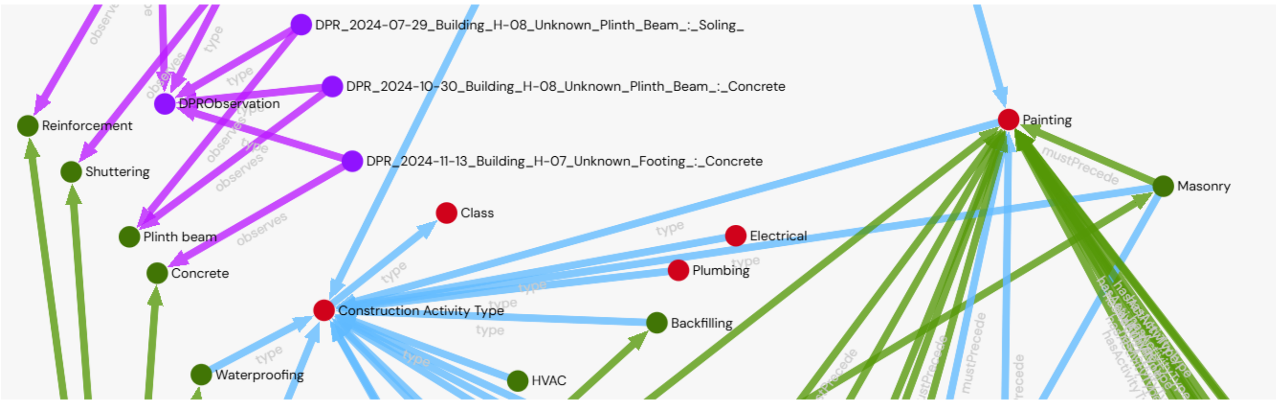

Figure 5 Knowledge Graph Evolution Across Pipeline Stages. Baseline nodes and edges are in red; schedule parsing-based nodes and edges are in green, DPR parsing-based nodes and edges are in violet.

invalid matches. This result is reflected in consistently higher median confidence values and the absence of constraint violations, even when operating on the DPR files. Of the 1000 activities, 769 alignment matches passed all constraint validation, and the remaining 231 activities were left unmatched, achieving zero logical conflicts in the inferred knowledge graph. Relative to the LLM baseline, the main gain is better confidence calibration rather than higher peak scores.

A key observation from the experiments is that system performance is driven by the ability to calibrate confidence. The LLM baseline occasionally produced higher peak confidence values, but these were not supported by symbolic validation. In contrast, the proposed approach is more robust, limiting the extreme confidence, since incorrect high-confidence updates can distort the KG for downstream applications.

The presence of 775 recorded interaction events with successful refinements suggests that embedding symbolic reasoning within the inference loop enables systematic reconciliation rather than post-hoc filtering. This behavior directly supports the effectiveness of a Type 6 neuro-symbolic integration as discussed earlier. The ablation study further reinforces this observation, showing that knowledge graph validation, vocabulary learning, and iterative refinement mechanisms account for nearly 70% of the system's performance contribution.

Error analysis shows that most strict UID failures were cross building or granularity confusions rather than semantic misunderstandings, indicating that the main remaining limitation is spatial disambiguation. Knowledge accumulation was substantial with KG growth dominated by precedence closure and 375 vocabulary mappings acquired during DPR processing. Validation on a single project remains a limitation, although the dataset spans 173 DPR files (i.e., 173 days of data) and 1000 activities, and the initial ontology is domain-general. The current validation does not stress-test contradictory schedules, leaving multi-project and synthetic conflict evaluation as future research directions. Overall, the findings indicate that the value of the proposed approach lies not in outperforming the neural baselines on isolated similarity metrics but in delivering a scalable, interpretable, and constraint-aware mechanism for schedule enrichment under realistic field conditions.

Conclusion and Future Work

This paper presented a bi-directional neuro-symbolic framework for automated construction schedule enrichment using daily progress reports. By embedding symbolic constraint reasoning within a large language model-driven inference loop, the proposed system enables semantically flexible, logically governed alignment between heterogeneous field documentation and formal schedules. Validation on a real construction project demonstrated that the framework achieves robust alignment performance, characterized by high median confidence, zero accepted constraint violations, and substantial growth of project-specific knowledge without manual ontology engineering.

Comparative baseline and ablation studies confirmed that the system's effectiveness arises from neuro-symbolic synergy rather than isolated neural or rule-based components. Symbolic validation and iterative refinement were shown to actively govern inference, preventing logically invalid schedule updates while allowing the language model to adapt to informal terminology and evolving field context. The learning trajectory analysis further indicated that the system transitions from rapid vocabulary acquisition to stable, conservative alignment behavior as domain knowledge matures, a desirable property for operational deployment.

Several directions for future work naturally follow from this study. First, while the current validation focused on a single project, extending the framework to multi-project datasets would enable systematic evaluation of generalizability across different schedule structures and documentation practices. Second, stress-testing the system with synthetic or adversarial contradictions, such as intentionally inconsistent precedence or delayed prerequisite activities, would provide deeper insight into its conflict-detection and resolution capabilities. Third, the current symbolic constraints rely on empirically defined thresholds. Adaptive or data-driven thresholds could further improve the robustness under varying data quality conditions. Finally, tighter integration with BIM may enable more explicit reasoning about spatial dependencies, thereby enhancing the system's ability to interpret fine-grained execution workflows.

Overall, the results suggest that the neuro-symbolic integration offers a promising and practical pathway towards scalable, interpretable, and optionally safe automation of construction schedule management. By balancing semantic adaptability with explicit domain constraints, the proposed framework establishes a foundation for future research on intelligent, learning-driven planning systems in construction.

Acknowledgements

The authors would like to thank Mr. Divyansh Gupta for his extensive support in data collection and coordination with the project field team. The authors also gratefully acknowledge the Indian Institute of Technology Bombay for facilitating access to project data and supporting this research.

References

Adekunle, P., Aigabvboa, C., Thwala, W., Akinradewo, O., & Oke, A. (2022). Challenges confronting construction information management. *Frontiers in Built Environment*, *8*, 1075674. doi: 10.3389/fbuil.2022.1075674

Al-Sinan, M. A., Bubshait, A. A., & Aljaroudi, Z. (2024). Generation of Construction Scheduling through Machine Learning and BIM: A Blueprint. *Buildings*, *14*(4), 934. doi: 10.3390/buildings14040934

Amer, F., Jung, Y., & Golparvar-Fard, M. (2021). Transformer machine learning language model for auto-alignment of long-term and short-term plans in construction. *Automation in Construction*, *132*, 103929. doi: 10.1016/j.autcon.2021.103929

Amer, F., Jung, Y., & Golparvar-Fard, M. (2023). Construction schedule augmentation with implicit dependency constraints and automated generation of lookahead plan revisions. *Automation in Construction*, *152*, 104896. doi: 10.1016/j.autcon.2023.104896

Cai, L., Yu, C., Kang, Y., Fu, Y., Zhang, H., & Zhao, Y. (2025). Practices, opportunities and challenges in the fusion of knowledge graphs and large language models. *Frontiers in Computer Science*, *7*, 1590632. doi: 10.3389/fcomp.2025.1590632

Golparvar-Fard, M., Peña-Mora, F., & Savarese, S. (2015). Automated Progress Monitoring Using Unordered Daily Construction Photographs and IFC-Based Building Information Models. *Journal of Computing in Civil Engineering*, *29*(1), 04014025. doi: 10.1061/(ASCE)CP.1943-5487.0000205

He, C., Yu, B., Liu, M., Guo, L., Tian, L., & Huang, J. (2024). Utilizing Large Language Models to Illustrate Constraints for Construction Planning. *Buildings*, *14*(8), 2511. doi: 10.3390/buildings14082511

Kautz, H. A. (2022). The third AI summer: AAAI Robert S. Engelmore Memorial Lecture. *AI Magazine*, *43*(1), 105–125. doi: 10.1002/aaai.12036

Li, H., Yang, R., Xu, S., Xiao, Y., & Zhao, H. (2024). Intelligent Checking Method for Construction Schemes via Fusion of Knowledge Graph and Large Language Models. *Buildings*, *14*(8), 2502. doi: 10.3390/buildings14082502

Olawale, Y., & Sun, M. (2015). Construction project control in the UK: Current practice, existing problems and recommendations for future improvement. *International Journal of Project Management*, *33*(3), 623–637. doi: 10.1016/j.ijproman.2014.10.003

Park, J., & Cho, Y. K. (2022). Point Cloud Information Modeling: Deep Learning–Based Automated Information Modeling Framework for Point Cloud Data. *Journal of Construction Engineering and Management*, *148*(2), 04021191. doi: 10.1061/(ASCE)CO.1943-7862.0002227

Project Management Institute. (2021). *A Guide to the Project Management Body of Knowledge (PMBOK® Guide) – Seventh Edition and The Standard for Project Management*. Project Management Institute.

Saka, A. B., & Chan, D. W. M. (2020). Adoption and implementation of building information modelling (BIM) in small and medium-sized enterprises (SMEs): A review and conceptualization. *Engineering, Construction and Architectural Management*, *28*(7), 1829–1862. doi: 10.1108/ECAM-06-2019-0332

Turkan, Y., Bosche, F., Haas, C. T., & Haas, R. (2012). Automated progress tracking using 4D schedule and 3D sensing technologies. *Automation in Construction, Planning Future Cities-Selected Papers from the 2010 eCAADe Conference*, *22*, 414–421. doi: 10.1016/j.autcon.2011.10.003

Wu, C., Li, X., Jiang, R., Guo, Y., Wang, J., & Yang, Z. (2023). Graph-based deep learning model for knowledge base completion in constraint management of construction projects. *Computer-Aided Civil and Infrastructure Engineering*, *38*(6), 702–719. doi: 10.1111/mice.12904

Wu, S., Yao, X., Liu, S., Liang, H., & Liu, Z. (2024). Domain Knowledge Graph Construction Methods of Construction Schedule in Steel Structure Projects. *2024 16th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, 39–44. doi: 10.1109/IHMSC62065.2024.00017