ITcon
www.itcon.org - Journal of Information Technology in Construction - ISSN 1874-4753

PERFORMANCE-DRIVEN CLAUSE PRIORITIZATION AND SMART CONTRACT SYNTHESIS FOR CONSTRUCTION CONTRACT GOVERNANCE: A PROOF-OF-CONCEPT FRAMEWORK AND PRELIMINARY EVALUATION

SUBMITTED: May 2026
PUBLISHED: September 2026
EDITOR: Robert Amor
DOI: 10.36680/j.itcon.2026.046

Erfan Moayyed, Ph.D. Candidate
M. E. Rinker Sr. School of Construction Management, University of Florida, Gainesville, FL, USA
https://orcid.org/0000-0003-4670-8453
moayyederfan@ufl.edu

Chimay J. Anumba, Ph.D.
College of Design, Construction and Planning, University of Florida, Gainesville, FL, USA
https://orcid.org/0000-0003-4583-5928
anumba@ufl.edu

SUMMARY: *Construction contracts govern payment, scheduling, change management, risk allocation, and dispute resolution, yet their scale and heterogeneity keep contract administration largely manual. Existing large language model (LLM) approaches extract and classify clauses but remain disconnected from project performance metrics, standard-form requirements, and executable contract logic. This paper proposes and preliminarily evaluates a retrieval-augmented contract intelligence framework that integrates clause extraction, semantic classification, KPI-aware importance ranking, standards alignment, discrepancy diagnostics, and constrained smart contract synthesis as a proof-of-concept pipeline. Clause importance is quantified by mapping contract language to four performance indicators (cost overrun impact, schedule delay impact, cash-flow adequacy, and dispute frequency); these indicators support a structured prioritization heuristic rather than an empirically validated predictive model. The framework was evaluated on 588 clauses from five executed contracts of a large public owner, spanning general contractor, construction manager, architect-engineer, commissioning, and design-build delivery. Moderate inter-annotator agreement (69.17%; Cohen's $\kappa = 0.52$) motivated a taxonomy revision from ten to eight categories, after which the domain-calibrated classifier achieved a macro-averaged F1 of 0.718 on a balanced validation set and 0.572 on the full corpus, whereas a chain-of-thought LLM prompt achieved 0.483 on the same balanced set. Clause rankings remained stable across alternative KPI weightings (Spearman $\rho > 0.98$). Preliminary findings indicate that contractual influence is concentrated in payment and approval provisions across delivery methods, and that governance risk in this corpus arises mainly through omission and weakening of provisions rather than direct contradiction of standard forms. Constrained template-based synthesis raised smart contract quality from 0.8/4.0 to 3.6/4.0 relative to unconstrained generation. These findings suggest that retrieval-augmented language models, combined with structured performance reasoning and domain-specific calibration, may support selective, performance-driven contract governance, and they motivate broader validation across owner types and contract families.*

KEYWORDS: *Retrieval-Augmented Generation, construction contract intelligence, clause-level performance reasoning, standards alignment, smart contract synthesis.*

REFERENCE: *Moayyed, E., & Anumba, C. J. (2026). Performance-driven clause prioritization and smart contract synthesis for construction contract governance: A proof-of-concept framework and preliminary evaluation. Journal of Information Technology in Construction (ITcon), 31, 1096-1124. https://doi.org/10.36680/j.itcon.2026.046*

1. INTRODUCTION

Construction projects are governed by extensive contract documents that define responsibilities, payment procedures, change management workflows, risk allocation mechanisms, and dispute resolution processes among participating organizations. These documents frequently exceed hundreds of pages and combine multiple agreement forms, general conditions, supplementary provisions, and technical specifications. As project delivery systems become more fragmented and contractual interfaces increase, contract administration has emerged as a critical determinant of project performance. Empirical studies consistently identify contractual ambiguity, delayed notices, inconsistent interpretation of obligations, and weak enforcement mechanisms as major contributors to cost overruns, schedule delays, cash flow instability, and disputes in construction projects (Cheung & Pang, 2013; Love et al., 2016; Yiu & Cheung, 2006).

Despite their central role, construction contracts are still administered largely through manual review, experience-driven judgment, and document-centric workflows. Contract managers must interpret complex legal language, track time-sensitive obligations, and align contractual provisions with evolving project conditions. These processes are labor-intensive and prone to oversight, especially under conditions of uncertainty such as material price volatility, supply chain disruptions, and accelerated delivery schedules (Mitkus & Mitkus, 2014; Zhao et al., 2022). The lack of systematic, data-driven support for contract interpretation and enforcement increases administrative friction and limits the ability of project teams to respond proactively to emerging risks.

Recent advances in natural language processing and large language models have created new opportunities for automating the analysis of legal and technical documents. Within construction informatics, recent studies have applied retrieval-augmented generation to information retrieval and question answering (Wu et al., 2025), embedding optimization and semantic chunking to improve contract question answering (Zhang & Ning, 2026), and fine-tuned language models for construction law article identification (Zhou et al., 2026). Semantic methods have also been developed for standard-form contract selection and suitability assessment (Elkhayat & Marzouk, 2022; Nasser et al., 2026). More broadly, large language models have demonstrated strong performance in clause extraction, document classification, semantic similarity, and obligation identification across legal domains (Holzenberger et al., 2021; Chalkidis & Kampas, 2019), and Retrieval-Augmented Generation further strengthens these capabilities by grounding model outputs in external knowledge sources, reducing hallucination and improving traceability of reasoning (Lewis et al., 2020; Izacard & Grave, 2021). While these advances significantly improve document understanding, they remain disconnected from project performance metrics, a reasonable consequence of their primary objective of general-purpose legal text processing rather than domain-specific governance analysis. To the authors' knowledge, none of these systems provides a mechanism for linking contract semantics to measurable project outcomes or prioritizing clauses according to their governance significance.

Within the construction domain, prior research on contract automation has focused on rule-based compliance checking, ontology-driven representations of contractual obligations, and conceptual models for blockchain-supported procurement and payment automation (Zhang & El-Diraby, 2012; Lee et al., 2016; Turk & Klinc, 2017). These studies provide important foundations, but they typically address isolated aspects of the contract lifecycle, a scoping choice that allowed each to achieve depth within a bounded problem (e.g., rule-based compliance checking, ontology representation, or procurement automation individually) rather than pursuing integration across the contract lifecycle. Existing approaches treat clause extraction, semantic analysis, standards alignment, and automation as largely independent problems and do not provide a unified reasoning framework that links contract semantics to project performance outcomes and executable governance mechanisms. Moreover, few studies provide quantitative validation connecting contract language to project performance indicators or assess the robustness of automated reasoning under alternative assumptions.

Three gaps emerge from this review. First, existing NLP and RAG-based systems classify and retrieve construction clauses but provide no mechanism to quantify which provisions matter most for project governance outcomes, leaving the relationship between contract language and performance impact unaddressed. Second, standards alignment between project-specific contracts and industry reference forms such as AIA A201 remains a largely manual process in the reviewed literature; to the authors' knowledge, no automated method has yet been reported that detects clause-level discrepancies with performance-weighted severity scoring linked to project outcomes. Third, smart contract synthesis for construction has relied on predefined logic or unconstrained generation, neither of which is grounded in the specific discrepancies and performance implications identified within a particular contract's provisions.

This paper proposes and preliminarily evaluates a Retrieval-Augmented Generation-based contract intelligence framework that addresses these gaps through a unified clause-level reasoning architecture. Specifically, the framework's KPI-aware importance modeling stage addresses the first gap by linking clause semantics to project performance dimensions; its standards alignment and discrepancy detection stage addresses the second gap through automated, severity-scored comparison against reference forms; and its constrained smart contract synthesis stage addresses the third gap by grounding executable logic in the specific discrepancies identified within each contract. Building on a system dynamics model that quantifies the relationship between contractual governance mechanisms and project performance indicators (Moayyed et al., 2026), the framework maps extracted clauses to performance dimensions, detects standards-based discrepancies, and generates constrained smart contract logic targeting identified governance deficiencies. The proof-of-concept evaluation uses five executed contracts from a large public owner, with a companion clause-level retrieval study on the same corpus providing additional methodological grounding (Moayyed & Anumba, 2026). This work's contribution is characterized as the systems integration of retrieval-augmented generation, clause classification, standards comparison, and template-based logic generation, each drawing on established techniques. The contribution lies in introducing a structured reasoning layer that connects these components, linking contract language to performance outcomes and execution logic within a single pipeline, enabling a transition from document-centric analysis to performance-driven contract governance.

This study is guided by the following research questions:

RQ1. How effectively can a Retrieval-Augmented Generation-based system extract, classify, and rank clauses across heterogeneous construction contract types according to their contribution to project performance indicators, and how robust are these rankings under alternative weighting assumptions?

RQ2. What patterns of contractual influence emerge from performance-oriented clause prioritization, and what do these patterns reveal about the governance structure of construction contracts?

RQ3. To what extent can automated standards alignment identify and characterize contractual discrepancies, and how can these discrepancies be operationalized into constrained smart contract logic?

This study makes four contributions to construction contract informatics. First, it provides preliminary evidence that contractual influence is highly concentrated: payment and approval provisions account for a disproportionate share of modeled governance relevance across delivery method types, introducing what is, to the authors' knowledge, an early clause-to-KPI reasoning layer in construction contract analytics; the underlying KPI weights are analytically assigned rather than empirically calibrated against observed project outcomes, so this layer is best understood as a structured prioritization heuristic rather than a validated predictive instrument. Second, it shows that governance risk in a public owner corpus arises primarily through omission and weakening of provisions rather than direct contradiction of standard forms, a finding with practical implications for how contract review and standards compliance should be prioritized. Third, it demonstrates that constrained template-based smart contract synthesis substantially outperforms unconstrained LLM generation, establishing a feasible contract-to-execution pathway grounded in real contract data and identified discrepancies. Fourth, it shows that zero-shot LLM classification underperforms domain-calibrated structured approaches for construction contract analysis, consistent with companion retrieval findings on the same corpus and confirming that domain-specific adaptation is necessary before LLM-based methods can replace structured approaches in this setting.

2. BACKGROUND AND RELATED WORK

2.1 Construction contract analysis and clause-level informatics

Construction contracts function as dynamic governance instruments that encode obligations, allocate risk, and regulate decision-making across organizational boundaries (Yiu & Cheung, 2006; Cheung & Pang, 2013). From a project management perspective, their effectiveness depends on how efficiently payment, scheduling, change management, and dispute resolution mechanisms are structured and enforced. As noted above, these administrative weaknesses are empirically linked to adverse project outcomes. Public-sector contracts are particularly complex, often incorporating owner-specific modifications to standard forms such as AIA A201 that create heterogeneous clause structures requiring continuous interpretation throughout execution.

Early construction informatics research addressed these challenges through rule-based and ontology-driven representations of contractual obligations (Zhang & El-Diraby, 2012; Lee et al., 2016). While these approaches enabled structured representations of contract concepts, they required extensive manual modeling and lacked scalability across heterogeneous contract forms. More recent work has applied NLP to automate clause extraction and semantic comparison: studies have examined semantic selection of standard-form contracts (Elkhayat & Marzouk, 2022), analysis of contract selection factors using NLP (Nasser et al., 2026), supervised extraction of payment and risk-allocation provisions (Zhao et al., 2022), and LLM-based structured data extraction from construction contracts (Moayyed & Anumba, 2024). Machine learning has also been applied to dispute-adjacent prediction tasks, such as forecasting construction contract dispute outcomes from case characteristics and root causes (Un et al., 2025), though this line of work operates at the level of predicting dispute resolution rather than linking clause-level contract language to governance performance. These approaches improve document understanding at the clause or category level but do not provide mechanisms to quantify clause importance or link individual provisions to downstream project performance outcomes. Clause-level granularity matters because contractual risk and control are implemented through individual provisions rather than entire documents, yet existing analytics tools do not prioritize clauses for managerial attention or automation based on governance significance.

Standards alignment between project-specific contracts and industry reference forms such as AIA A201 has received limited attention in the informatics literature. Prior studies focus on conceptual alignment frameworks or manual checklist-based reviews. Automated discrepancy detection at the clause level with performance-weighted severity scoring remains unexplored, limiting the ability to prioritize corrective actions or assess the governance consequences of specific deviations from reference standards.

2.2 Large language models and retrieval-augmented generation in construction informatics

Transformer-based language models have significantly advanced legal text understanding, enabling clause extraction, semantic similarity assessment, and obligation detection across diverse legal domains (Chalkidis & Kampas, 2019; Holzenberger et al., 2021). These studies address general legal NLP tasks rather than construction contracts specifically; their relevance to the present setting is therefore inferential, resting on the assumption that strong performance on general legal text extends to the more specialized structural and drafting conventions of construction contracts. Retrieval-Augmented Generation strengthens these capabilities by conditioning model outputs on retrieved external sources, improving factual consistency and traceability of reasoning (Lewis et al., 2020; Izacard & Grave, 2021).

Within construction informatics, recent studies have applied RAG to information retrieval and question answering in construction management (Wu et al., 2025), embedding optimization and semantic chunking to improve contract question answering accuracy (Zhang & Ning, 2026), and fine-tuned language models for construction law article identification in dispute resolution (Zhou et al., 2026). A systematic review of LLM-based approaches for document-to-smart contract transformation identifies key research opportunities and implementation challenges in this space (Moayyed et al., 2025). An evaluation of clause-level retrieval methods on the same corpus used in the present study found that structured retrieval approaches outperform naive semantic baselines for construction contracts, suggesting that domain-specific constraints are necessary for reliable performance in this setting (Moayyed & Anumba, 2026).

Despite these advances, existing RAG and LLM-based systems are designed primarily for document understanding and question answering. None connects retrieval or generation outputs to project performance indicators, standards compliance diagnostics, or executable governance logic. The question of which clauses matter most for project outcomes, and how identified discrepancies should inform automation decisions, remains unaddressed. Whether zero-shot LLM classification can substitute for domain-calibrated structured approaches in this setting is evaluated empirically in Section 4.2.

2.3 Smart contract automation and executable governance logic

Blockchain-enabled smart contracts have been proposed as mechanisms to automate payment processing, milestone approvals, and compliance workflows in construction (Turk & Klinc, 2017; Li et al., 2019; Wang et al., 2020). Empirical evaluations have demonstrated their potential for improving supply chain visibility and payment integrity (Hamledari & Fischer, 2021). System dynamics modeling has further quantified how automation-enabled governance mechanisms can reduce contractual frictions and moderate project performance under macroeconomic volatility (Moayyed et al., 2026), establishing quantitative linkages between governance process delays and performance outcomes across cost, schedule, cash flow, and dispute dimensions.

Despite this potential, most smart contract research in construction remains detached from real contract documents. Existing approaches typically assume predefined contractual logic and do not address the translation of natural language clauses into executable code. Few studies identify which provisions should be automated, validate automation logic against actual contract language, or connect synthesis outputs to performance implications. Without systematic clause selection, standards validation, and performance-impact assessment, smart contract implementations risk being technically feasible but operationally misaligned with project governance needs.

Table 1: Comparison of prior work and proposed approach.

Study	Focus	Level of analysis	Key capability	Limitation	Contribution of this study
Zhang and El-Diraby (2012); Lee et al. (2016)	Ontology / rule-based compliance	Clause / semantic	Structured representation and rule-based validation	Limited scalability; no performance linkage; no prioritization	Introduces KPI-based clause importance and performance-aware reasoning
Zhao et al. (2022); Moayyed and Anumba (2024)	NLP / LLM-based contract analysis	Document / clause	Clause extraction and classification	No linkage to project performance; no standards integration	Extends extraction to performance-grounded clause prioritization
Lewis et al. (2020) (RAG)	Retrieval-augmented reasoning	Document-level	Context-aware retrieval and generation	Not domain-specific; no contract performance interpretation	Applies RAG within a performance-aware contract reasoning framework
AIA (2017) standards-based approaches	Standards compliance	Document / checklist	Benchmarking against standard forms	Manual evaluation; no severity or impact quantification	Automated discrepancy detection with KPI-aware severity assessment
Elkhayat and Marzouk (2022)	Selecting feasible standard forms of construction contracts using text analysis	Contract / form level	Semantic comparison of standard forms to support contract selection	Supports contract-form choice but not clause-level discrepancy detection, severity scoring, or KPI linkage	Moves from contract-form selection to clause-level standards alignment and severity-aware discrepancy analysis
Nasser et al. (2026)	Semantic analysis of construction contract selection factors using NLP	Contract / factor level	NLP-based identification and analysis of contract selection factors	Focuses on contract selection criteria rather than clause prioritization, standards diagnostics, or executable logic	Extends semantic contract analysis toward clause-level governance relevance and remediation-oriented automation

Study	Focus	Level of analysis	Key capability	Limitation	Contribution of this study
Turk and Klinc (2017); Li et al. (2019); Wang et al. (2020)	Blockchain / smart contracts	System / conceptual	Contract automation concepts	No systematic translation from contract text; no clause prioritization	Clause-driven smart contract synthesis linked to performance impact
Hamledari and Fischer (2021)	Impact of blockchain and smart contracts on construction supply chain visibility	System / process level	Evaluates blockchain and smart contract potential for transparency and visibility	Focuses on process-level visibility benefits; does not derive executable logic from natural-language contracts	Connects real contract clauses and detected discrepancies to structured, parameterized smart contract logic
Moayyed et al. (2025)	LLM review	Conceptual	Synthesizes document-to-smart contract methods	No implementation or validation	Operationalizes concepts into a working system
Moayyed and Anumba (2024)	Data extraction	Clause	Extracts structured contract data	No KPI mapping; no standards alignment	Extends to KPI reasoning and discrepancy analysis
Wu et al. (2025)	RAG-driven information retrieval and question answering in construction management	Document / query level	Retrieval-grounded question answering and information access in construction documents	Focuses on retrieval and QA; does not rank clauses by governance importance or connect outputs to project KPIs	Extends RAG from information access to performance-grounded clause prioritization and executable governance logic
Zhang and Ning (2026)	Contract question answering through embedding optimization and semantic chunking	Document / chunk level	Improves contract QA through better embeddings and chunking strategies	Improves retrieval quality but does not perform standards alignment, KPI reasoning, or automation prioritization	Builds on clause-level semantic retrieval by linking clauses to KPI impact, standards discrepancies, and structured automation
Zhou et al. (2026)	Article-level law identification using fine-tuned LLMs in construction disputes	Article / legal reference level	Fine-tuned legal article identification for construction case relevance	Targets legal article retrieval rather than contract clause governance, performance impact, or remediation logic	Extends legal-semantic identification toward contract-specific governance reasoning and clause-to-execution mapping
Moayyed et al. (2026)	System dynamics	System-level	Defines KPI relationships	No clause-level linkage	Bridges clause-level semantics to system-level KPIs
This Study	Performance-grounded contract intelligence system	Clause → KPI → execution	Clause-level reasoning, standards-aware diagnostics, robust KPI prioritization, and constrained logic synthesis	—	First empirically assessed clause-to-performance-to-execution framework for construction contract governance

Taken together, research across these three streams has advanced individual aspects of construction contract analysis but has not established a unified reasoning pathway connecting contract language to project performance outcomes and executable governance logic. The present study addresses this by introducing a clause-level reasoning architecture that integrates these dimensions as a proof-of-concept, evaluated on real executed contracts. Table 1 summarizes the positioning of this work relative to prior studies.

Building on this positioning, the remainder of this paper addresses the three gaps identified in Section 1 through a three-stage pipeline architecture. The first stage, clause extraction and KPI-aware importance modeling (Sections 3.2-3.3), addresses the absence of a mechanism linking contract language to project performance outcomes. The second stage, standards alignment and discrepancy detection (Section 3.4), addresses the lack of automated, severity-scored comparison against reference forms such as AIA A201. The third stage, constrained smart contract synthesis (Section 3.5), addresses the reliance on unconstrained generation by grounding executable logic in the specific discrepancies identified within each contract. The remainder of this section details the methodology underlying each stage.

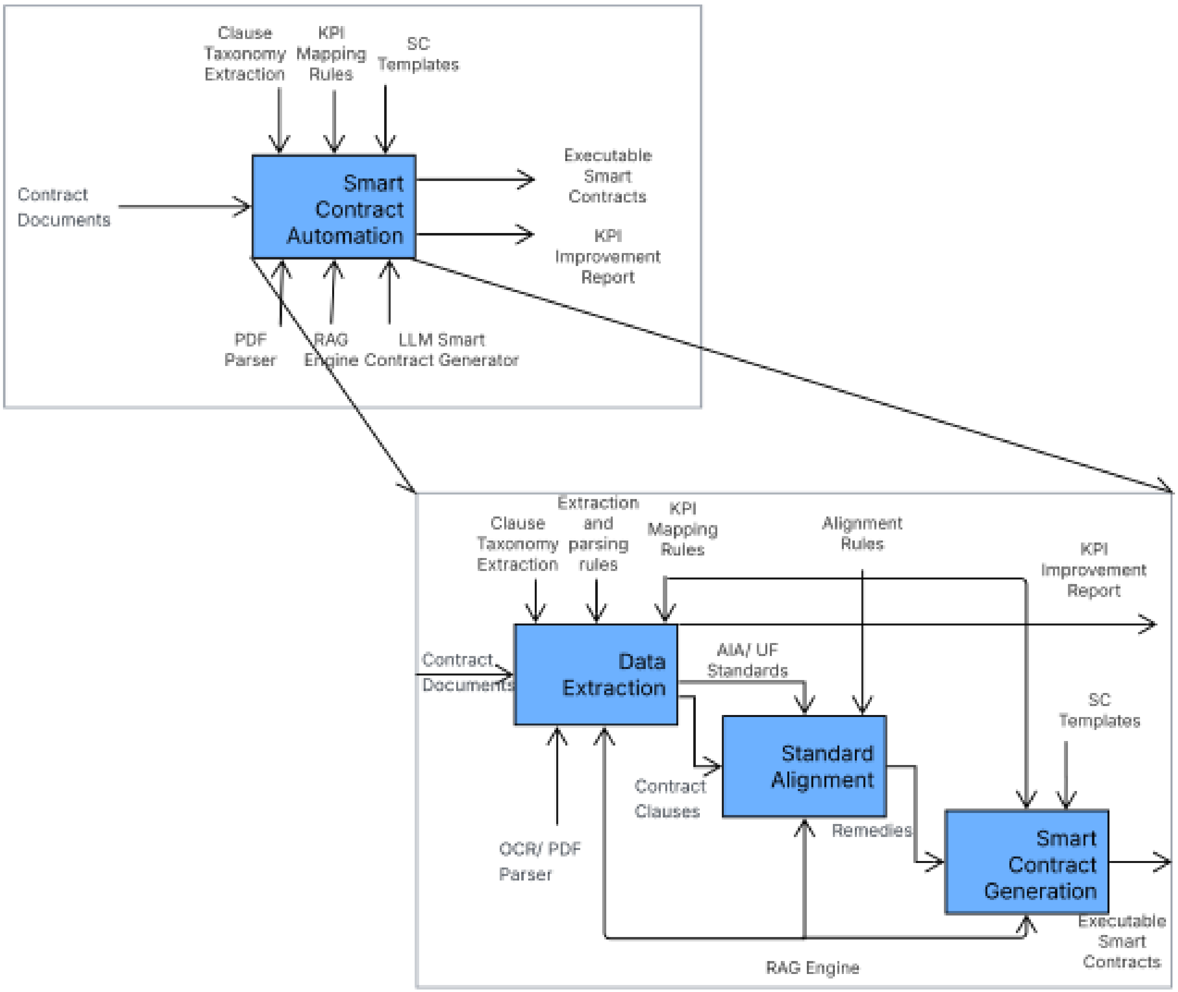


Figure 1: Overview of the Retrieval-Augmented contract intelligence framework showing clause extraction, KPI reasoning, standards alignment, and smart contract synthesis stages.

3. METHODS

This study proposes a Retrieval-Augmented Generation-based contract intelligence framework for clause-level analysis and structured smart contract synthesis. The framework is organized as three sequential stages: (1) clause extraction and semantic classification, (2) KPI-aware importance modeling and standards-aligned discrepancy analysis, and (3) impact estimation and constrained clause-to-logic synthesis. The overall workflow is shown in Figure 1.

At a high level, raw contract documents are first transformed into structured clause representations. These clauses are then evaluated through clause classification, KPI linkage, and standards alignment. Finally, validated clause-level outputs are translated into structured logic templates and aggregated into project-level impact indicators.

3.1 Contract corpus and data description

The framework was evaluated using five executed construction contracts obtained from a large public owner. The corpus includes contracts representing multiple delivery roles and procurement methods, specifically Construction Manager, General Contractor, Architect-Engineer, Commissioning, and Design-Build agreements. These documents reflect common public-sector contracting practices in a U.S. context and include both primary agreement forms and incorporated general terms and conditions.

All contracts were provided in digital text format. After preprocessing and clause segmentation, a total of 588 individual clauses were extracted across the five contracts. Each clause served as the fundamental analytical unit in the framework. Table 2 summarizes the corpus, including contract type, total pages, number of extracted clauses, and dominant clause categories. Contract identifiers are anonymized for confidentiality.

Table 2: Contract corpus.

Contract id	Contract type	Contract role	Total pages	Extracted clauses	Dominant clause categories
XX-XXX_DB	Design-Build	Design-Builder	41	214	Schedule, Change Management, Risk Allocation
XX-XXX_GC_Executed-1	Design-Bid-Build	General Contractor	58	167	Payment, Schedule, Disputes and Claims
XX-XXX_SHCC_AE	Design Services	Architect / Engineer	30	96	Submittals and Approvals, Professional Obligations
XX-XXX_SHCC_CM	Construction Management	Construction Manager	31	64	Schedule Coordination, Risk Allocation
XX-XXX_SHCC_Cx	Commissioning	Commissioning Agent	14	47	Compliance, Verification and Testing

3.2 Clause extraction and semantic classification

Classification performance was evaluated using a manually labeled subset of 120 clauses randomly sampled from the corpus. Precision, recall, and F1 scores were computed for each category, along with a macro-averaged F1 score to provide an overall measure independent of class frequency. Table 3 reports classification performance across the ten clause categories.

To assess the reliability of the ground truth labels, inter-annotator agreement was computed on the same subset, resulting in an agreement rate of 69.17% and a Cohen's κ of 0.52, indicating moderate agreement. Disagreement was concentrated in boundary-prone categories such as submittals and approvals provisions, general conditions, and notices, reflecting the inherent semantic ambiguity of legal language rather than purely model-related limitations. Detailed annotation procedures and agreement statistics are provided in Appendix A and Appendix B.

Performance varies across categories, with lower F1 scores observed in semantically overlapping provisions such as notices, disputes and claims, and termination. These categories frequently involve cross-referenced obligations and multifunctional language, making strict categorical separation challenging even for human annotators. Accordingly, the classification component should be interpreted as providing a structured basis for downstream reasoning under conditions of semantic ambiguity, rather than error-free categorization.

Table 3: Classification performance of the clause categorization.

Category	Precision	Recall	F1	Error rate (1−F1)
schedule	0.732	0.612	0.667	0.333
change_orders	0.714	0.588	0.645	0.355
payment	0.619	0.639	0.629	0.371
risk_allocation	0.719	0.561	0.630	0.370
submittals_approvals	0.656	0.553	0.600	0.400
other	0.753	0.416	0.536	0.464
general_conditions	0.627	0.396	0.486	0.514
disputes_claims	0.375	0.500	0.429	0.571

To contextualize performance, the proposed hybrid classifier was compared against baseline approaches, including a rule-based keyword classifier and an embedding-based nearest-neighbor classifier. The hybrid model substantially outperformed both baselines, demonstrating the value of combining structured domain constraints with contextual semantic reasoning. Full comparative results are reported in Appendix B.

Because subsequent stages of the framework depend on classification outputs, robustness was further evaluated through stress testing. This included exclusion of low-performing categories and confidence-weighted adjustments based on category-level F1 scores. These tests were designed to assess the extent to which classification uncertainty propagates through downstream KPI modeling and discrepancy analysis. Results indicate that core qualitative patterns remain stable despite variations in classification accuracy. Detailed robustness results are provided in Appendix B.

Figure 2 illustrates the clause extraction and classification pipeline, including the interaction between embedding, retrieval, and language model inference.

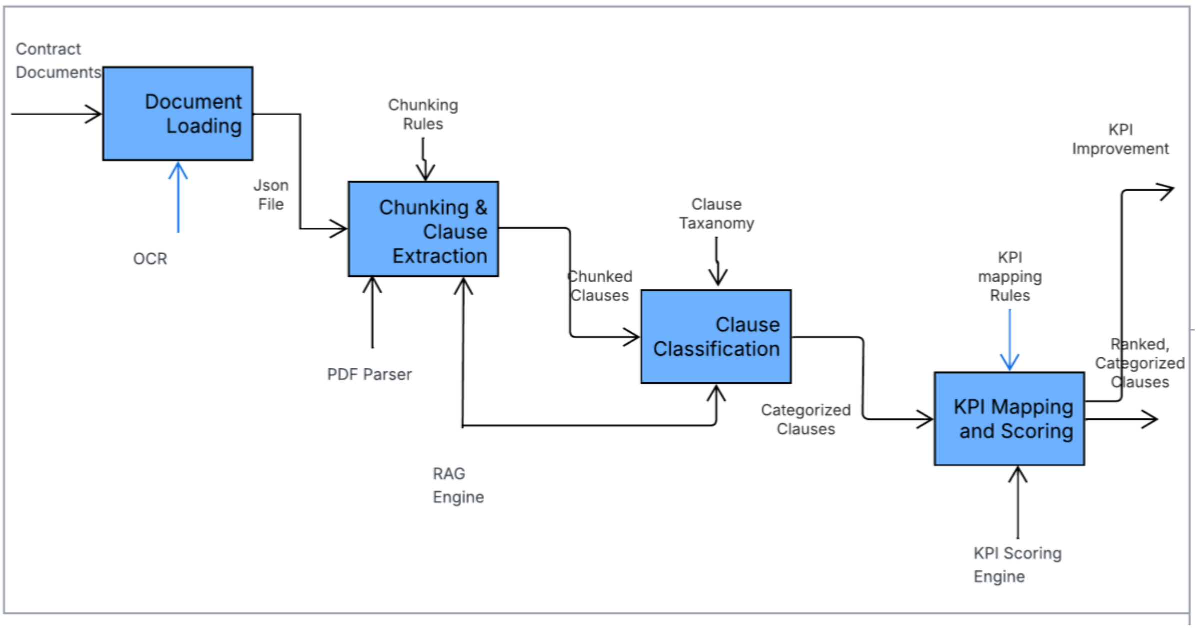


Figure 2: Clause extraction and KPI attribution process model.

3.3 KPI definition and clause importance modeling

To quantify the contribution of contract clauses to project performance, four Key Performance Indicators (KPIs) are defined: cost overrun impact (COI), schedule delay impact (SDI), cash-flow adequacy (CFA), and dispute frequency (DF). These KPIs represent core dimensions of project outcomes that are strongly influenced by contractual governance mechanisms. Each clause is mapped to one or more KPIs based on its semantic content and functional role within the contract.

These four KPIs were selected because they are the project outcome dimensions most consistently identified in the construction project management literature as being shaped by contractual governance mechanisms (Cheung & Pang, 2013; Love et al., 2016; Mitkus & Mitkus, 2014). Cost and schedule outcomes are the two indicators most widely used to define project success in this literature (Chan et al., 2004; Flyvbjerg et al., 2004), motivating the inclusion of COI and SDI; cash-flow adequacy and dispute frequency were added as mediating governance dimensions through which payment and administrative frictions are known to propagate into cost and schedule impacts. Alternative or more granular KPI sets, for example, separating rework, safety, or quality-related outcomes into additional indicators, were not evaluated in this proof-of-concept; the four-KPI structure represents a deliberately parsimonious starting point rather than an exhaustive taxonomy of project performance, and expert elicitation of an expanded or alternative KPI set is identified as a direction for the companion validation study. Accordingly, the clause-to-KPI mappings and importance rankings that follow should be read as a structured, literature-grounded prioritization heuristic rather than an empirically validated predictive model of project outcomes.

The linkage strength between clause c and KPI k is estimated using a hybrid formulation that combines rule-based domain knowledge with language model inference:

$$l_{c,k} \;=\; \alpha\, l_{c,k}^{\text{rule}} \;+\; (1-\alpha)\, l_{c,k}^{\text{LLM}} \tag{1}$$

Equation 1 expresses the generalized blending formulation; the present prototype implementation approximates it through a rule-prioritized decision hierarchy rather than computing a continuous alpha-mixture directly, as detailed below. The correspondence between the generalized formulation and this implemented approximation is evaluated via the sensitivity analysis reported later in this subsection.

where $l_{c,k}$ denotes the estimated influence of clause c on KPI k, $l_{c,k}^{\text{rule}}$ represents the rule-based KPI linkage derived from the clause taxonomy, $l_{c,k}^{\text{LLM}}$ captures the semantic inference based on contextual interpretation. The blending parameter $\alpha \in [0,1]$ controls the relative influence of deterministic taxonomy mappings and semantic inference. In the present prototype implementation, linkage assignment was operationalized through a rule-prioritized decision hierarchy that first used explicit clause KPI annotations when available and otherwise inferred KPI relevance from category mappings. To examine consistency with the generalized formulation, an ex post sensitivity analysis evaluated alternative α-mixtures. Results showed that across α values of 0.25, 0.50, and 0.75, Spearman rank correlations with the baseline linkage assignments ranged from 0.923 to 0.935, with top-10 clause overlap ranging from 0.80 to 0.90, indicating that the prioritization structure is stable across alternative blending assumptions.

Each clause is associated with a KPI linkage vector:

$$L_c \;=\; \left(I_{c,\text{COI}}, I_{c,\text{SDI}}, I_{c,\text{CFA}}, I_{c,\text{DF}}\right) \tag{2}$$

These linkage values are normalized to ensure bounded contributions across performance dimensions. The formulation builds on prior system dynamics modeling of construction performance (Moayyed et al., 2026), extending it by introducing a clause-level mapping that connects contractual provisions to system-level performance variables. where $l_{c,k} \in [0,1]$ represents the estimated influence of clause c on KPI k. These linkage values are derived using a hybrid rule-based and language-model reasoning process that evaluates the semantic role of each clause and allocates fractional influence across the four KPIs. The rule-based component maps clause categories to expected performance effects based on domain knowledge of construction contract administration, while the language model evaluates semantic context and distributes fractional influence across KPIs when clauses affect multiple performance dimensions. The overall importance score of clause c is then computed as:

$$I(c) \;=\; \sum_{k} W_k I_{c,k} \tag{3}$$

where $I(c)$ represents the overall importance score of clause c, W_k denotes the project-level weight assigned to KPI k, and $I_{c,k}$ represents the normalized linkage strength between clause c and KPI k. The clause–KPI linkage formulation is intended as a structured reasoning mechanism that operationalizes domain knowledge about how contractual provisions affect project governance, rather than as an empirically calibrated predictive model of project outcomes. The weighting and linkage structure therefore supports relative prioritization and sensitivity analysis, but should not be interpreted as establishing causal effect sizes from observed project performance data.

$$\sum_k w_k = 1 \tag{4}$$

In this study, the baseline KPI weight vector is defined as w = (0.35, 0.30, 0.20, 0.15) for cost overrun impact (COI), schedule delay impact (SDI), cash-flow adequacy (CFA), and dispute frequency (DF), respectively. This ordering reflects widely documented performance priorities in construction project management research. Cost and schedule outcomes are consistently treated as primary success indicators across empirical studies of infrastructure project performance (Chan et al., 2004; Flyvbjerg et al., 2004; Love et al., 2016), while cash-flow adequacy and dispute frequency represent secondary governance dimensions whose effects are mediated through cost and schedule pathways (Cheung & Pang, 2013; Moayyed et al., 2026).

Because the specification of KPI weights introduces potential subjectivity, robustness analysis was conducted using alternative weighting scenarios, including equal weighting and stress cases emphasizing individual KPIs. Clause importance rankings remained highly stable across all tested configurations, with Spearman rank correlation coefficients exceeding 0.98. This indicates that the model captures structurally influential clauses rather than artifacts of a specific weighting choice. Full robustness results are reported in Appendix C.

Figure 3 illustrates the computational logic of the clause importance model. Extracted clauses are first categorized and mapped to KPI linkage vectors using the hybrid formulation described above. The linkage values are then aggregated using the defined KPI weights to compute clause-level importance scores. These scores enable ranking of clauses within and across contracts and provide structured inputs for downstream modules, including standards alignment, discrepancy analysis, and smart contract synthesis.

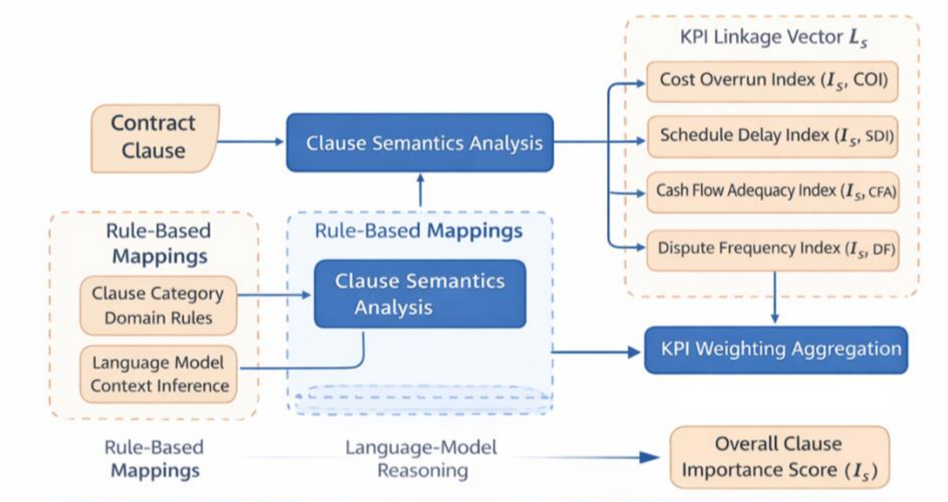


Figure 3: Logic used to quantify clause importance with respect to project performance.

Table 4 presents the distribution of average KPI contributions across clause categories. To ensure comparability, aggregated contributions are normalized by the number of clauses within each category, thereby reflecting per-clause impact rather than total category contribution. The results indicate that payment-related clauses exhibit the highest average impact across all KPIs, particularly for cost overrun and cash-flow adequacy, highlighting their central role in project financial performance.

Submittals and approvals, as well as disputes and claims clauses, also demonstrate relatively high per-clause contributions, reflecting their influence on coordination delays and dispute escalation pathways. In contrast, categories such as schedule and general conditions show lower per-clause impact despite their prevalence, suggesting that many provisions in these categories are descriptive or procedural rather than performance-critical.

*Table 4: Distribution of clause importance values across KPI categories**

Clause category	Clauses (n)	Avg COI (%)	Avg SDI (%)	Avg CFA (%)	Avg DF (%)	Avg overall (%)
payment	137	0.734	0.634	0.435	0.082	0.549
schedule	27	0.148	0.097	0.085	0.046	0.106
disputes_claims	12	0.127	0.360	0.072	0.180	0.195
submittals_approvals	91	0.430	0.047	0.246	0.184	0.242
general_conditions	97	0.268	0.019	0.027	0.115	0.121
other*	224	0.031	0.013	0.017	0.013	0.020

*Categories with fewer than 10 clauses are omitted for display clarity; their average KPI contributions were below 0.05 across all dimensions.

These findings indicate that contractual influence is not uniformly distributed but concentrated in a subset of high-impact provisions. Accordingly, the results support a selective automation strategy that prioritizes clauses with high impact density rather than attempting comprehensive contract digitization.

3.4 Standards alignment and discrepancy detection

The third stage of the framework evaluates contract clauses against reference standards to identify potential deviations, inconsistencies, and areas of misalignment. Although explicit references to specific standard forms (e.g., AIA A201) are not always present, the analyzed contracts consistently incorporate provisions derived from widely adopted industry standards. For each clause, the framework retrieves the most relevant standard sections using semantic similarity search over a curated standards repository. The repository consists of hand-authored files providing paraphrased summaries of standard-form provisions relevant to each contract-type family analyzed in this study (Construction Manager, General Contractor, Architect-Engineer, Commissioning, and Design-Build), drawing on AIA A201, B101, C203, and A141 as reference points; provisions were paraphrased rather than reproduced verbatim, consistent with copyright restrictions on AIA-published text. Repository content is segmented by markdown header and paragraph boundaries and indexed using sentence-transformer embeddings for similarity-based retrieval. Construction of the repository prioritized payment-related provisions, reflecting the concentration of governance relevance in payment clauses identified in Section 4.3; consequently, dedicated standards content was not developed for the remaining clause categories (schedule, notices, change orders, risk allocation, submittals and approvals, disputes and claims, and termination) at the time of the analysis reported here. For clauses in these categories, standards retrieval returns no matching content, and the alignment process instead relies on the underlying language model's general knowledge of construction contract practice rather than direct comparison against indexed reference text. This scope limitation is a methodological constraint of the present proof-of-concept; expanding the repository to provide direct standards coverage across all clause categories is identified as a priority for the companion validation study.

Table 5: Discrepancy severity scoring scheme.

Discrepancy type	Description	Severity score
Missing provision	Clause required by standards is absent	1.00
Conflicting provision	Clause contradicts recommended standards	0.80
Weaker provision	Clause partially satisfies standards but with weaker obligations	0.60
Ambiguous provision	Clause wording is vague or subject to interpretation	0.40
Minor deviation	Small deviation from standard practice	0.20

Clause-to-standard alignment is then performed using a Retrieval-Augmented Generation (RAG) process that compares contractual provisions with normative requirements. This process combines vector-based retrieval with prompt-constrained reasoning, enabling the identification of differences such as missing elements, weakened obligations, or ambiguous phrasing relative to the reference standard. Detected discrepancies are categorized into predefined types, including missing provisions, conflicting provisions, weaker provisions, and ambiguous provisions. Each discrepancy is assigned a normalized severity score $s_c \in [0,1]$, representing the expected magnitude of governance risk introduced by the discrepancy.

The severity values reported in Table 5 are designed to represent increasing levels of potential governance risk associated with contractual deviations. The scoring scheme preserves monotonic scaling across discrepancy categories. No prior construction-informatics literature specifies numerical severity weights for this discrepancy taxonomy, so the specific values were calibrated exploratorily rather than adopted from an external source: candidate spacings, including equal 0.25 increments and a non-linear geometric spacing, were each propagated through the KPI impact aggregation model (Equation 5) and evaluated against two criteria – monotonic ordering of resulting clause-level impact rankings across discrepancy types, and absence of discontinuities in the normalized project-level score (Section 3.5) as severity varies. The linear 0.20-increment scale reported in Table 5 was retained because it satisfied both criteria while preserving distinguishable mid-range severity levels that the tested non-linear alternatives compressed. This calibration should be understood as a structured design choice supporting stable relative prioritization, not an empirically derived risk-weighting scheme validated against observed project outcomes.

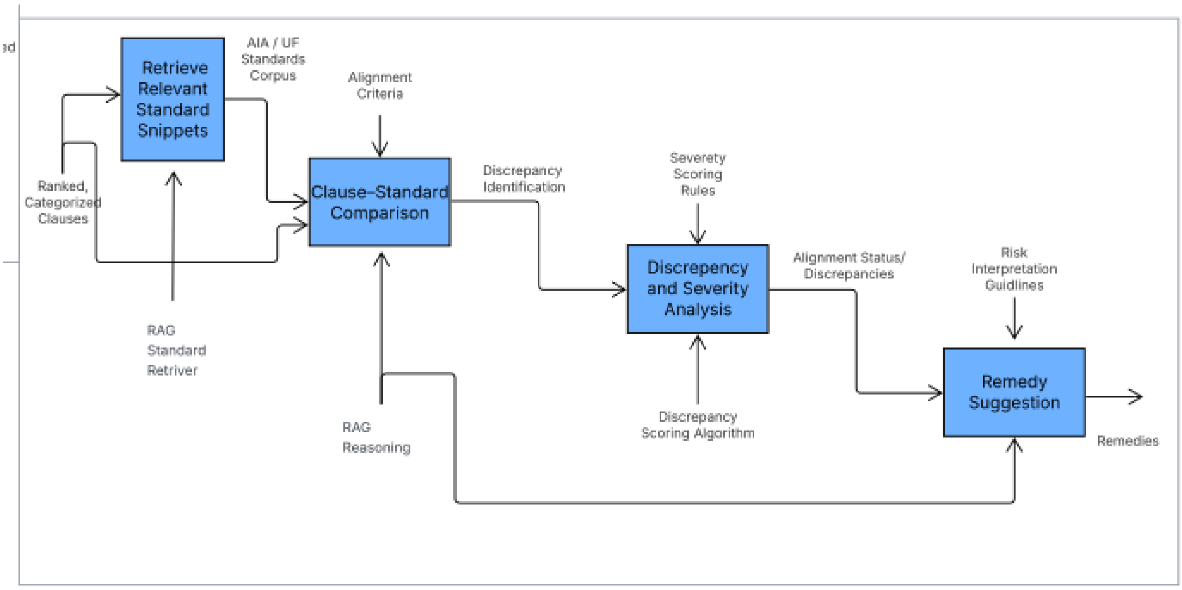


Figure 4: Standards alignment and discrepancy detection process.

Importantly, discrepancy identification is performed at the clause level and does not necessarily imply that an obligation is entirely absent at the contract level. In construction contracts, provisions are frequently distributed across multiple clauses or expressed in alternative forms. As a result, a clause-level classification of "missing" should be interpreted as the absence of a specific standardized formulation within that clause, rather than the definitive absence of the obligation across the entire contract. To address this limitation, a second-pass contract-context audit was introduced. For each discrepancy initially classified as "missing," the framework searches other clauses within the same contract to identify potential relocated or distributed coverage of the corresponding obligation. Based on this analysis, discrepancies are reclassified into three categories: confirmed missing, possibly relocated, and uncertain. This audit is implemented under both strict and moderate thresholds to assess sensitivity to evidence criteria. The full methodology and results of this audit are provided in Appendix D.

Figure 4 illustrates the standards alignment and discrepancy detection process, including retrieval, comparison, and classification stages.

Discrepancy types are detected and defined here (Table 5); the observed distribution of discrepancy types and severity across the corpus is reported as an empirical finding in Section 4.4 (Table 8).

3.5 KPI impact estimation and smart contract compilation

Clause-level discrepancies identified during the standards alignment stage are translated into potential performance improvements through a remediation impact model. For each clause c and KPI k, the expected improvement associated with correcting a detected discrepancy is estimated as:

$$\Delta\mathrm{KPI}_{c,k} = l_{c,k} \cdot s_c \cdot r_c \tag{5}$$

where $\Delta\mathrm{KPI}_{c,k}$ represents the modeled improvement in KPI k resulting from remediation of the discrepancy associated with clause c, $l_{c,k}$ is the clause–KPI linkage strength defined in Section 3.3, s_c is the discrepancy severity score, and $r_c \in [0,1]$ is a remediation effectiveness coefficient representing the proportion of potential benefit realized after corrective action. This formulation is intended as a structured scenario-analysis mechanism rather than a predictive estimator of realized project outcomes. It provides a consistent method for comparing the relative value of addressing different contractual deficiencies under explicit assumptions.

Clause-level KPI improvements are aggregated using the KPI weighting structure introduced in Section 3.3. To avoid amplification effects when multiple discrepancies affect the same KPI simultaneously, guardrails are applied to cap individual clause contributions within predefined ranges. Project-level impact is reported using a normalized bounded score defined as the ratio of modeled cumulative gains to the maximum attainable gains under the same linkage structure:

$$NS = \frac{\sum_{c \in C} I(c)}{\sum_{c \in C} I_c^{\max}} \tag{6}$$

where $I_c^{\max}$ represents the maximum attainable contribution for clause c under full severity and full remediation. This normalization improves interpretability by expressing improvement as a fraction of modeled achievable benefit, rather than as an unbounded cumulative index. Because this normalized score is a novel construct without established benchmark values in prior construction contract literature, it is intended to be interpreted relative to its own bounds, 0% indicating no realized benefit from remediation and 100% indicating that every identified discrepancy achieves its maximum modeled, severity-weighted contribution under full remediation, and relative to the sensitivity scenarios reported alongside it, rather than against an external threshold.

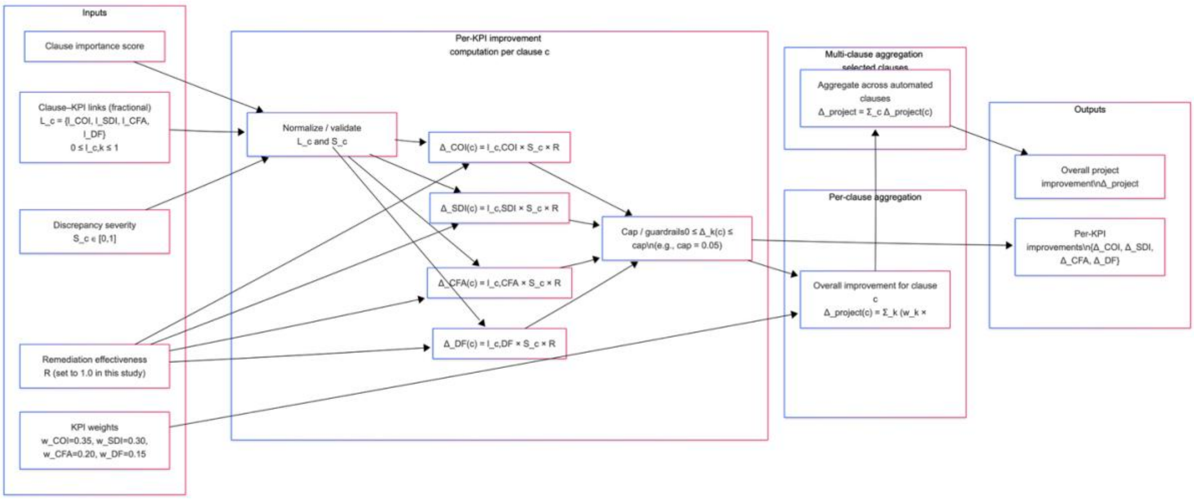


Figure 5: Mathematical structure of the KPI impact aggregation.

Baseline analysis was conducted using r_c = 1.0, representing full remediation effectiveness. Because this assumption may be optimistic in practice, additional sensitivity analyses were performed for r_c = 0.75 and r_c = 0.50. Rankings of high-priority clauses remained highly stable across scenarios (Spearman correlations ≈ 0.999; complete top-10 overlap), indicating that lower remediation effectiveness primarily reduces impact magnitude rather than altering prioritization conclusions. Full results are reported in Appendix C.

Figure 5 summarizes the computational logic of KPI impact aggregation, from clause-level remediation estimates to normalized project-level improvement scores.

To ensure numerical correctness and reproducibility, a batch validation script recomputed clause-level and project-level values under multiple severity and remediation scenarios. The validator enforced range constraints, zero-case invariants, and formula consistency across all 588 clauses. These procedures confirm internal computational consistency of the framework, although they do not constitute empirical validation against realized project outcomes.

Smart contract logic synthesis is implemented as a constrained compilation step. Instead of unrestricted text generation, the system maps validated clause discrepancies and recommended remedies to predefined executable logic templates. Templates cover common governance functions such as payment deadlines, milestone certification, late-payment interest, notice periods, change-order approvals, and retention release conditions. Parameter values (e.g., due dates, percentages, actors, thresholds) are extracted from clause text and inserted into the corresponding templates. Smart contract synthesis quality was evaluated using two complementary procedures. First, a rubric-based internal validation was performed on 10 representative clause–remedy–logic cases using four binary criteria: semantic consistency, remedy alignment, structural validity, and condition–action completeness. These four criteria were designed to track the natural stages of clause-to-logic translation – whether the generated logic preserves the clause's meaning, whether it reflects the intended correction, whether the resulting rule structure is internally coherent, and whether it is executable – rather than adopted from an established evaluation framework. To the authors' knowledge, no published rubric specifically targets clause-to-smart-contract synthesis quality; the criteria draw loosely on completeness and correctness dimensions used in broader legal-NLP evaluation practice but should be understood as an exploratory, author-defined instrument rather than a validated psychometric or field-standard rubric.

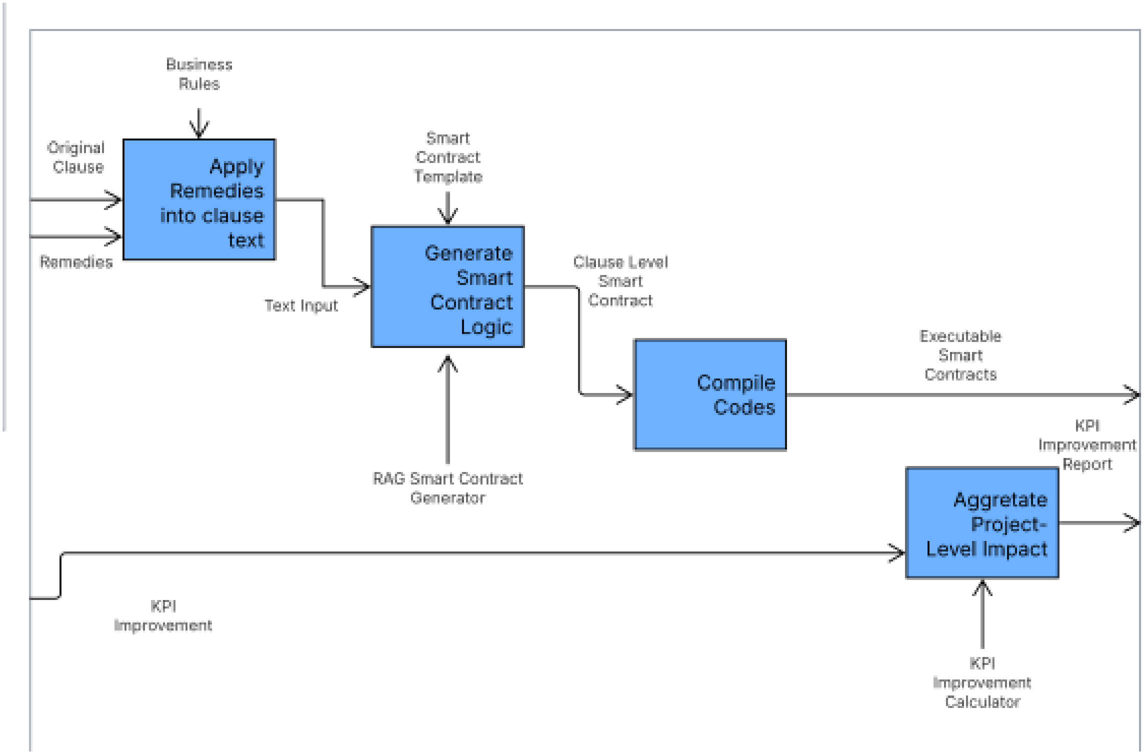


Figure 6: Transformation from validated clause representations to smart contract logic.

This template-constrained approach substantially improved output quality relative to an unconstrained baseline. In a sample-based validation exercise, the average synthesis quality score increased from 0.8/4.0 to 3.6/4.0, with major gains in semantic consistency, remedy alignment, and structural validity. Full validation details are reported in Appendix E. Figure 6 illustrates the transformation from validated clause representations to structured smart contract logic templates.

4. RESULTS

4.1 Contract corpus and clause statistics

Results below are reported for the same five-contract, 588-clause corpus described in Section 3.1. Because all contracts originated from a single public owner, the findings should be interpreted as initial evidence rather than statistically generalizable industry-wide estimates; broader validation across owners, jurisdictions, and contract families remains a priority for future research.

4.2 Clause extraction and classification performance

The Classification performance was evaluated iteratively across three stages: baseline assessment using the original ten-category taxonomy, taxonomy revision motivated by inter-annotator analysis, and comparison against a chain-of-thought LLM prompt to assess the value of domain-specific calibration relative to zero-shot generative classification.

Performance was initially assessed on a manually labeled validation subset of 120 clauses randomly sampled from the full corpus, representing approximately 20% of all extracted clauses. Inter-annotator agreement on this subset yielded an agreement rate of 69.17% and a Cohen's kappa of 0.52, indicating moderate agreement. Disagreement was concentrated in boundary-prone categories including submittals and approvals, general conditions, and notices, suggesting that part of the observed classification difficulty reflects genuine semantic ambiguity in contractual language rather than purely algorithmic limitations. Detailed annotation procedures and agreement statistics are provided in Appendix A.

The original ten-category taxonomy achieved a macro-averaged F1 of 0.550 on the full five-contract corpus. Stronger performance was achieved for structurally explicit categories such as schedule (0.667), change orders (0.645), risk allocation (0.630), and payment (0.629). Lower performance was observed for procedurally entangled categories including notices (0.421), disputes and claims (0.429), and termination (0.455). Analysis of inter-annotator disagreements revealed that confusion was concentrated precisely in these categories, where provisions function as prerequisites or consequences of other contractual mechanisms: notices serve as procedural prerequisites to claims, and termination provisions operate as consequences of risk allocation failures. This pattern motivated a taxonomy revision from ten to eight categories, merging notices into disputes and claims and termination into risk allocation to reflect functional dependencies rather than enforce boundaries that neither human annotators nor the classifier could reliably maintain.

On the full five-contract corpus, the taxonomy revision improved macro-averaged F1 from 0.550 to 0.572, a consistent gain of 2.2 percentage points attributable to cleaner category boundaries. To enable a more representative per-category assessment, a balanced validation set of 141 clauses was constructed by supplementing the original 120-clause set with targeted annotations from underrepresented categories. Annotation candidates were identified through two methods: distant supervision from relevance judgments in a companion clause-level retrieval study on the same corpus (Moayyed & Anumba, 2026), which yielded 15 high-confidence weak labels across disputes and claims, schedule, and submittals and approvals; and keyword-based search targeting change orders and risk allocation provisions. On this balanced validation set, the domain-calibrated classifier achieved a macro-averaged F1 of 0.718 across well-represented categories, with disputes and claims reaching F1 of 1.000 at a support of 8 and schedule reaching F1 of 1.000 at a support of 10. Table 6 summarizes classification performance across all three evaluation stages.

The 0.718 figure is not directly comparable to the 0.550 full-corpus baseline, as it reflects a more balanced evaluation set and a revised taxonomy applied to a different clause sample. The full-corpus metric of 0.572 represents the consistent apples-to-apples comparison across the entire five-contract dataset.

To assess whether a generative approach could match or exceed the domain-calibrated classifier, a chain-of-thought LLM prompt incorporating category-specific few-shot examples and disambiguation rules was evaluated on the same balanced validation set. The LLM prompt achieved a macro-averaged F1 of 0.483, substantially below the domain-calibrated classifier on the same set. Per-category examination shows that the LLM systematically over-predicted general conditions (precision 0.14), pulling payment clauses (recall 0.71) and other provisions (recall 0.35) into the broad catch-all category when uncertain. This behavior reflects the absence of domain-specific training: without calibration on construction contract data, the LLM defaults to the broadest available category rather than the most functionally specific one. This pattern is consistent with findings from the companion retrieval study on the same corpus, which showed that structured approaches outperform naive semantic baselines for construction contracts (Moayyed & Anumba, 2026), and confirms that domain-specific adaptation is necessary before generative classification can reliably replace structured approaches in this setting.

Two categories, change orders and risk allocation, showed sparse representation in the validation set despite targeted annotation efforts. Corpus analysis using the companion retrieval study revealed that this single public-owner corpus assigns 81.5% of clauses to a broad general conditions category, indicating that change-order and risk-allocation provisions are predominantly embedded within general conditions sections rather than appearing as standalone clause types. This is a structural characteristic of the corpus, not a classification failure. These categories are retained in the taxonomy for extensibility across owner types and contract families, but are reported separately as sparse categories in the evaluation to ensure accurate macro-F1 computation.

The proposed hybrid classifier substantially outperformed two lightweight baselines: a rule-based keyword classifier (macro-F1 = 0.136) and an embedding-based nearest-neighbor classifier (macro-F1 = 0.147). Robustness testing under two stress scenarios, exclusion of low-performing categories and confidence-weighted adjustments, confirmed that core qualitative findings remained stable across both conditions. Payment clauses consistently dominated top-ranked positions and cash-flow adequacy remained the most prominent KPI pattern, with ranking similarity maintained at Spearman correlation values up to 0.98. Full baseline results and robustness analyses are provided in Appendix B.

Table 6: Model performance comparison.

Approach	Evaluation set	Categories	Macro-F1
Baseline (original classifier)	Full corpus	10-category	0.550
Taxonomy revision	Full corpus	8-category	0.572
Taxonomy revision	Balanced validation (141 clauses)	8-category	0.718
LLM prompt v2 (chain-of-thought)	Balanced validation (141 clauses)	8-category	0.483

4.3 Clause importance and KPI impact analysis

Following clause extraction and classification, each clause was evaluated in terms of its relative importance to overall project governance performance. Importance was defined as the extent to which a clause is linked to key project outcomes, operationalized through four Key Performance Indicators (KPIs): cost overrun impact (COI), schedule delay impact (SDI), cash-flow adequacy (CFA), and dispute frequency (DF). Each clause was associated with one or more KPIs based on its functional role within the contract, such as payment timing, notice procedures, change management, certification requirements, or dispute resolution. Clause-level importance scores were then computed by aggregating normalized KPI linkage values using the weighting framework described in Section 3.3. This formulation allows clauses affecting multiple performance dimensions to be distinguished from those with narrower administrative scope. The resulting scores enable direct comparison of provisions within and across contracts independent of document length or clause count. Table 7 presents representative high-ranking clauses across the analyzed contracts based on the computed importance scores. Contract identifiers are anonymized for confidentiality.

Across the five contracts, clauses governing payment certification, invoice processing, change-order authorization, time extensions, and notice requirements consistently ranked among the most influential. These provisions are closely associated with liquidity continuity, schedule control, and dispute prevention, explaining their dominant contribution within the prioritization model. Table 8 reports only the top five ranked clauses per contract; whether non-payment clauses appear within the broader top-ten distribution, and the full shape of that distribution across all ranked clauses, are not assessed here and are identified as a direction for more detailed reporting in future work. The rankings also reveal meaningful variation across contract types. Payment-related clauses were most prominent in General Contractor and Construction Manager agreements, where financial administration and progress payments are central governance functions. In contrast, Architect-Engineer agreements exhibited higher-ranked clauses related to approvals, certifications, and administrative coordination, which are more closely linked to schedule reliability and dispute avoidance. Design-Build contracts displayed a more distributed profile, with high-ranking clauses spanning payment, scope definition, and risk allocation.

Table 7: The highest-ranked clauses for each contract based on the computed importance scores (Clause identifiers are abbreviated section-level labels derived from anonymized contract records).

Contract	Clause ID	Category	Rank	Dominant KPI
XX-XXX_DB	6.3.4	Payment	1	COI
XX-XXX_DB	7.2	Payment	2	COI
XX-XXX_DB	8	Payment	3	COI
XX-XXX_DB	7	Payment	4	COI
XX-XXX_DB	7.3	Payment	5	COI
XX-XXX_GC_Executed-1	5.2	Payment	1	COI
XX-XXX_GC_Executed-1	5.3	Payment	2	COI
XX-XXX_GC_Executed-1	3.3.3	Payment	3	COI
XX-XXX_GC_Executed-1	4.1.6	Payment	4	COI
XX-XXX_GC_Executed-1	30	Payment	5	COI
XX-XXX_SHCC_CM	24.5	Payment	1	COI
XX-XXX_SHCC_CM	3.9.1.1	Payment	2	COI
XX-XXX_SHCC_CM	2.3.5	Payment	3	COI
XX-XXX_SHCC_CM	1.3	Payment	4	COI
XX-XXX_SHCC_CM	80.0	Payment	5	COI
XX-XXX_SHCC_AE	2.8.8	Payment	1	COI
XX-XXX_SHCC_AE	8.4	Payment	2	COI
XX-XXX_SHCC_AE	10.1.4.1	Payment	3	COI
XX-XXX_SHCC_AE	10.2	Payment	4	COI
XX-XXX_SHCC_AE	10.3	Payment	5	COI
XX-XXX_SHCC_Cx	5.2	Payment	1	COI
XX-XXX_SHCC_Cx	70.5	Payment	2	COI
XX-XXX_SHCC_Cx	5.1	Payment	3	COI
XX-XXX_SHCC_Cx	5.5	Payment	4	COI
XX-XXX_SHCC_Cx	6.4	Payment	5	COI

Sensitivity analysis of the α blending parameter across values of 0.25, 0.50, and 0.75 produced Spearman rank correlations of 0.923 to 0.935 with baseline rankings and top-10 clause overlap of 0.80 to 0.90. Payment clauses remained the dominant category under all tested configurations. This stability indicates that the concentration of governance influence in payment provisions is a structural characteristic of the corpus rather than an artifact of the specific rule-LLM blending ratio.

Additional robustness tests were performed to assess the effect of classification uncertainty. Under both exclusion and confidence-weighted stress scenarios, payment clauses continued to dominate top-ranked positions, and cash-flow adequacy remained the most prominent KPI pattern. Although absolute score magnitudes declined modestly, the overall ranking structure remained substantively consistent. Full robustness results are reported in Appendix C. These findings provide preliminary evidence that contractual influence is highly concentrated rather than uniformly distributed across the five analyzed contracts. A relatively small subset of payment and approval provisions consistently accounted for a disproportionate share of modeled governance relevance regardless of delivery method type. This concentration pattern, stable across alternative KPI weighting assumptions and classification stress scenarios, suggests that the distribution of governance significance within this corpus is structurally skewed toward financial administration and certification provisions rather than broadly shared across clause types; whether this pattern generalizes to construction contracts beyond this single public owner remains an open empirical question. This has practical implications for contract review and selective automation: the highest-leverage provisions can be identified systematically rather than through exhaustive document review, and automation resources can be directed toward provisions with the greatest modeled performance impact.

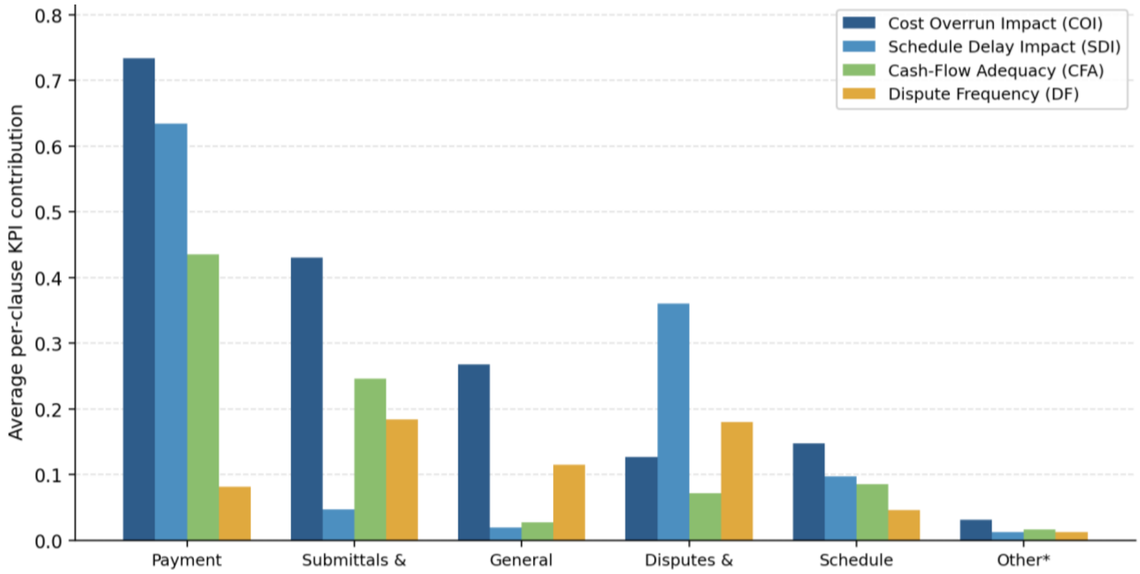


Figure 7: Average KPI contribution by clause category (payment, submittals and approvals, general conditions, disputes and claims, schedule, other), derived from Table 4. Payment clauses show the highest average contribution across cost overrun impact and schedule delay impact, consistent with the concentration finding discussed above.

4.4 Standards-based discrepancy analysis and severity patterns

Following the identification of high-priority clauses in Section 4.3, a standards-based alignment analysis was conducted to evaluate the extent to which contractual provisions diverge from commonly adopted industry practices. Each clause was compared against AIA-derived General Conditions using the proposed Retrieval-Augmented Generation pipeline, enabling clause-level detection of potentially missing, weaker, conflicting, or ambiguous provisions relative to reference formulations.

The observed distribution of discrepancy types and severity across the corpus is summarized in Table 8. Missing-type discrepancies constitute the largest category at the clause level, with 343 findings and an average severity score of 0.72. These findings were concentrated in provisions associated with payment timing, certification procedures, owner obligations, and administrative controls. Within the KPI framework, cash-flow adequacy (CFA) emerged as the dominant affected dimension for this discrepancy class.

Weaker provisions represented the second most common discrepancy type, with 35 findings and an average severity of 0.69. These clauses typically contained obligations that were present in principle but reduced in specificity, timing certainty, or enforceability relative to reference standards. Common examples included diluted payment certification language and less explicit approval deadlines. Similar to missing-type findings, weaker provisions were most strongly associated with cash-flow adequacy.

*Table 8: Discrepancy types and severity distributions. **

Discrepancy type	Clauses	Avg severity	Dominant KPI	High severity share (%)
Missing	343	0.72	CFA	100.0
Weaker	35	0.69	CFA	97.1
Ambiguous	12	0.65	CFA	100.0

*Conflicting and minor deviation discrepancy types were not identified in the analyzed corpus, consistent with the public owner's practice of deriving contract language from AIA standards rather than directly contradicting them.

Ambiguous provisions were less frequent, with 12 identified instances, but still exhibited elevated severity levels. Ambiguity commonly arose from undefined responsibilities, conditional phrasing, or fragmented cross-references that reduced interpretability. Although fewer in number, such clauses may increase governance friction and dispute exposure, particularly in payment-related workflows. Importantly, clause-level "missing" findings should not automatically be interpreted as definitive contract-level omissions. Contractual obligations are often distributed across multiple sections or expressed through alternative drafting structures. To address this issue, a second-pass contract-context audit was performed (Appendix D), reclassifying initially missing findings into confirmed missing, possibly relocated, and uncertain categories under strict and moderate evidence thresholds.

The consistent emergence of cash-flow adequacy as the dominant KPI across all three discrepancy types warrants interpretation. Payment-related clauses constitute the largest clause category in this corpus and carry the highest CFA linkage values. When standards alignment identifies a discrepancy in a provision that governs payment timing, certification, or owner obligations, the KPI impact model naturally attributes that discrepancy primarily to CFA. This pattern reflects a genuine structural characteristic of the analyzed contracts: the provisions most commonly deviating from AIA reference standards are those governing financial flows and payment administration. Whether this pattern holds across owner types and jurisdictions is an empirical question for the larger-scale validation study.

Under the strict setting, 19 findings remained confirmed missing, 98 were classified as possibly relocated, and 228 remained uncertain. Under the moderate setting, 19 remained confirmed missing, 161 were classified as possibly relocated, and 165 remained uncertain. These results support a cautious interpretation of initial discrepancy counts while preserving the broader signal that payment and administrative obligations are the most common loci of standards misalignment.

When interpreted jointly with the clause importance rankings, the discrepancy analysis suggests that a relatively small subset of high-priority clauses also exhibits elevated alignment risk. This concentration effect provides a structured basis for prioritizing standards review, remediation, and downstream automation efforts toward provisions with the greatest expected governance relevance.

4.5 Smart contract logic synthesis and projected KPI improvements

Building on the clause importance rankings and standards-based discrepancy analysis, the final stage of the framework translates validated contractual insights into structured smart contract logic and estimates their potential governance value through KPI-oriented scenario analysis. Figure 8 illustrates the user-facing interface of the smart contract synthesis module, which integrates clause-level classification, standards alignment outcomes, and KPI prioritization within a unified workflow.

Rather than generating code directly from raw contract text, the system synthesizes executable logic from two structured inputs: (i) baseline contractual obligations extracted from high-priority clauses, and (ii) remedial provisions derived from detected discrepancies relative to reference standards. This architecture reduces hallucination risk and preserves traceability between legal text, identified issues, and generated automation logic.

For each selected clause, the system first constructs baseline control logic representing the original contractual intent, including triggers, conditions, deadlines, approvals, and required actions. Where a discrepancy is identified, a supplementary remedial logic layer is introduced to address missing, weaker, or ambiguous provisions. Typical generated functions include automated notice deadlines, milestone payment releases, late-payment interest calculations, certification workflows, escalation triggers, retention release conditions, and change-order approvals.

The resulting logic, therefore, represents an augmented operational interpretation of the original clause rather than a replacement of the governing legal text.

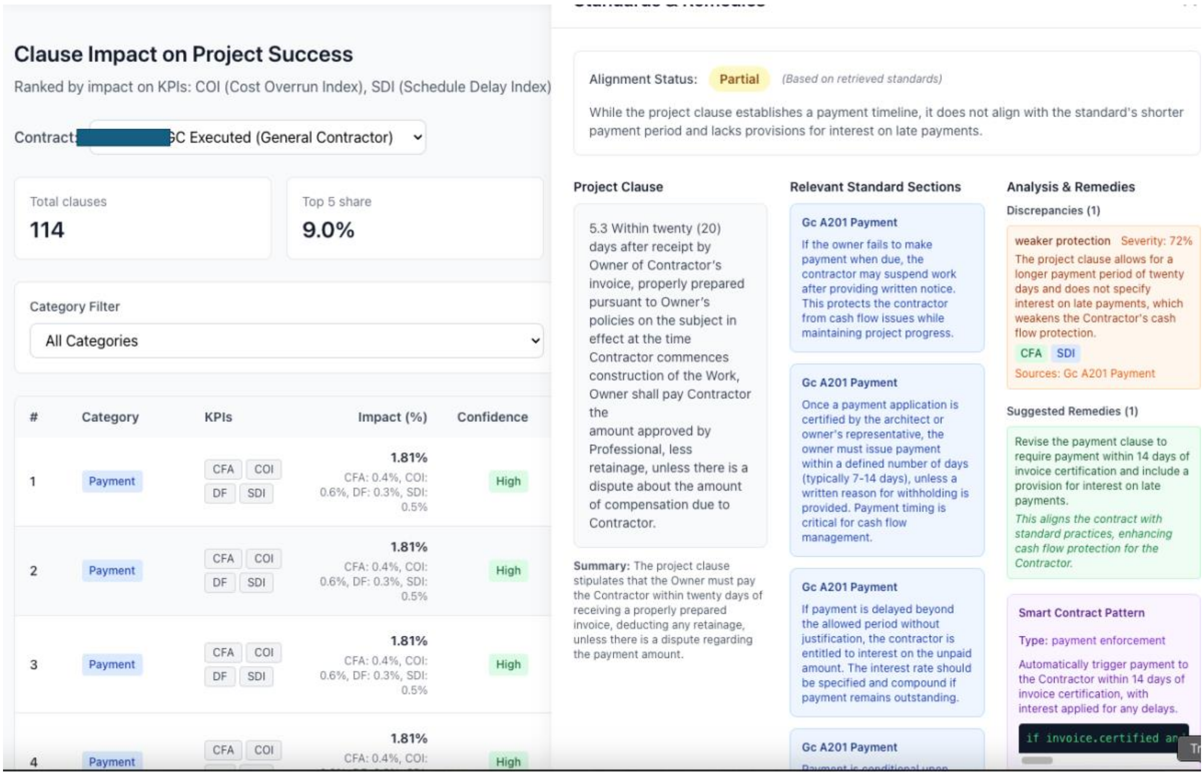
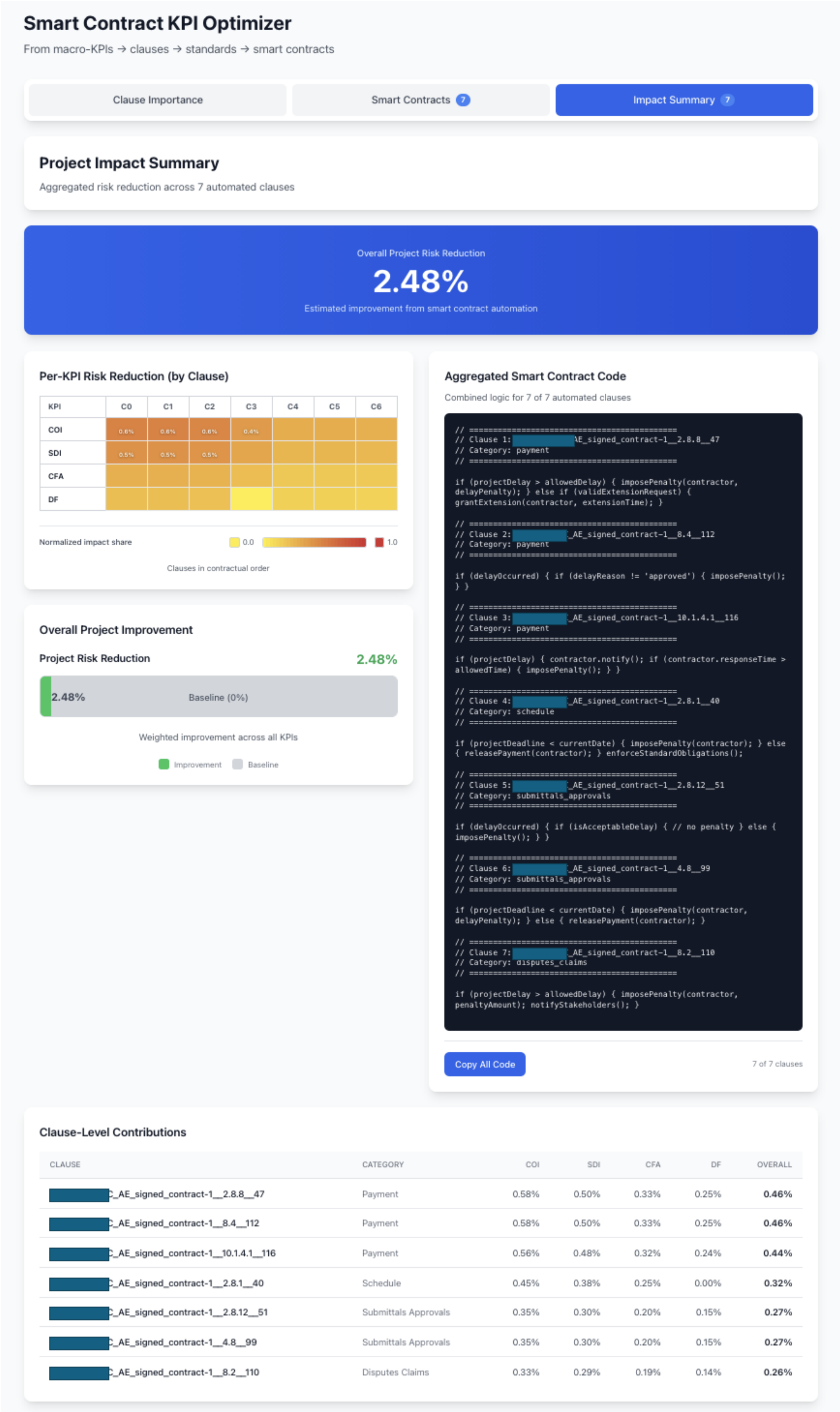


Figure 8: User-facing interface of the smart contract synthesis module.

To estimate potential project-level implications of these interventions, the framework applies the remediation impact model introduced in Section 3.5. Clause-level gains are aggregated into a normalized bounded project score representing the proportion of modeled achievable improvement under the same linkage structure. This bounded formulation avoids over-interpretation of cumulative indices and supports transparent comparison across scenarios. Under the baseline remediation effectiveness assumption (r_c = 1.0), the normalized project-level score reached 68.13% of modeled attainable improvement. This indicates that, under full remediation effectiveness, the modeled scenario captures roughly two-thirds of the maximum severity-weighted benefit theoretically available across all identified discrepancies; because no external benchmark exists for this construct, its principal interpretive value lies in comparison with the lower-effectiveness scenarios reported immediately below. Sensitivity analysis showed predictable attenuation under lower remediation effectiveness assumptions, declining to 52.28% for r_c = 0.75 and 35.88% for r_c = 0.50. Importantly, prioritization rankings remained highly stable across all scenarios (Spearman ≈ 0.999; complete top-10 overlap), indicating that remediation uncertainty primarily affects magnitude rather than clause selection. Full results are reported in Appendix C.

At the KPI level, payment timing, certification, and owner-obligation clauses generated the largest projected gains in cost overrun reduction and cash-flow adequacy. Notice, coordination, and procedural compliance clauses were more strongly associated with schedule reliability and dispute mitigation. Clauses with low importance scores or without detected discrepancies exhibited minimal projected gains, indicating that the framework selectively targets provisions with measurable governance leverage rather than recommending indiscriminate automation. Smart contract synthesis quality was evaluated through a sample-based rubric assessing semantic consistency, remedy alignment, structural validity, and condition-action completeness. The original unconstrained generation approach achieved an average score of 0.8/4.0. After introducing constrained template-based synthesis with parameter extraction, the average score increased to 3.6/4.0, with substantial gains in semantic fidelity and executable structure. Full validation details are reported in Appendix E. Figure 9 presents an example workflow from clause diagnosis to synthesized logic output.


Smart Contract KPI Optimizer
From macro-KPIs → clauses → standards → smart contracts
Clause Importance
Smart Contracts 7
Impact Summary 7
Project Impact Summary
Aggregated risk reduction across 7 automated clauses
Overall Project Risk Reduction
2.48%
Estimated improvement from smart contract automation
Per-KPI Risk Reduction (by Clause)
KPI C0 C1 C2 C3 C4 C5 C6
COI 0.6% 0.6% 0.6% 0.4%
SDI 0.5% 0.5% 0.5%
CFA
DF
Normalized impact share 0.0 1.0
Clauses in contractual order
Overall Project Improvement
Project Risk Reduction 2.48%
2.48% Baseline (0%)
Weighted improvement across all KPIs
Improvement Baseline
Aggregated Smart Contract Code
Combined logic for 7 of 7 automated clauses
// ==
// Clause 1: [redacted]AE_signed_contract-1__2.8.8__47
// Category: payment
// ==
if (projectDelay > allowedDelay) { imposePenalty(contractor, delayPenalty); } else if (validExtensionRequest) { grantExtension(contractor, extensionTime); }
// ==
// Clause 2: [redacted]_AE_signed_contract-1__8.4__112
// Category: payment
// ==
if (delayOccurred) { if (delayReason != 'approved') { imposePenalty(); } }
// ==
// Clause 3: [redacted]_AE_signed_contract-1__10.1.4.1__116
// Category: payment
// ==
if (projectDelay) { contractor.notify(); if (contractor.responseTime > allowedTime) { imposePenalty(); } }
// ==
// Clause 4: [redacted]_AE_signed_contract-1__2.8.1__40
// Category: schedule
// ==
if (projectDeadline < currentDate) { imposePenalty(contractor); } else { releasePayment(contractor); } enforceStandardObligations();
// ==
// Clause 5: [redacted]_AE_signed_contract-1__2.8.12__51
// Category: submittals_approvals
// ==
if (delayOccurred) { if (isAcceptableDelay) { // no penalty } else { imposePenalty(); } }
// ==
// Clause 6: [redacted]_AE_signed_contract-1__4.8__99
// Category: submittals_approvals
// ==
if (projectDeadline < currentDate) { imposePenalty(contractor, delayPenalty); } else { releasePayment(contractor); }
// ==
// Clause 7: [redacted]_AE_signed_contract-1__8.2__110
// Category: disputes_claims
// ==
if (projectDelay > allowedDelay) { imposePenalty(contractor, penaltyAmount); notifyStakeholders(); }
Copy All Code
7 of 7 clauses
Clause-Level Contributions
CLAUSE CATEGORY COI SDI CFA DF OVERALL
[redacted]_AE_signed_contract-1__2.8.8__47 Payment 0.58% 0.50% 0.33% 0.25% 0.46%
[redacted]_AE_signed_contract-1__8.4__112 Payment 0.58% 0.50% 0.33% 0.25% 0.46%
[redacted]_AE_signed_contract-1__10.1.4.1__116 Payment 0.56% 0.48% 0.32% 0.24% 0.44%
[redacted]_AE_signed_contract-1__2.8.1__40 Schedule 0.45% 0.38% 0.25% 0.00% 0.32%
[redacted]_AE_signed_contract-1__2.8.12__51 Submittals Approvals 0.35% 0.30% 0.20% 0.15% 0.27%
[redacted]_AE_signed_contract-1__4.8__99 Submittals Approvals 0.35% 0.30% 0.20% 0.15% 0.27%
[redacted]_AE_signed_contract-1__8.2__110 Disputes Claims 0.33% 0.29% 0.19% 0.14% 0.26%

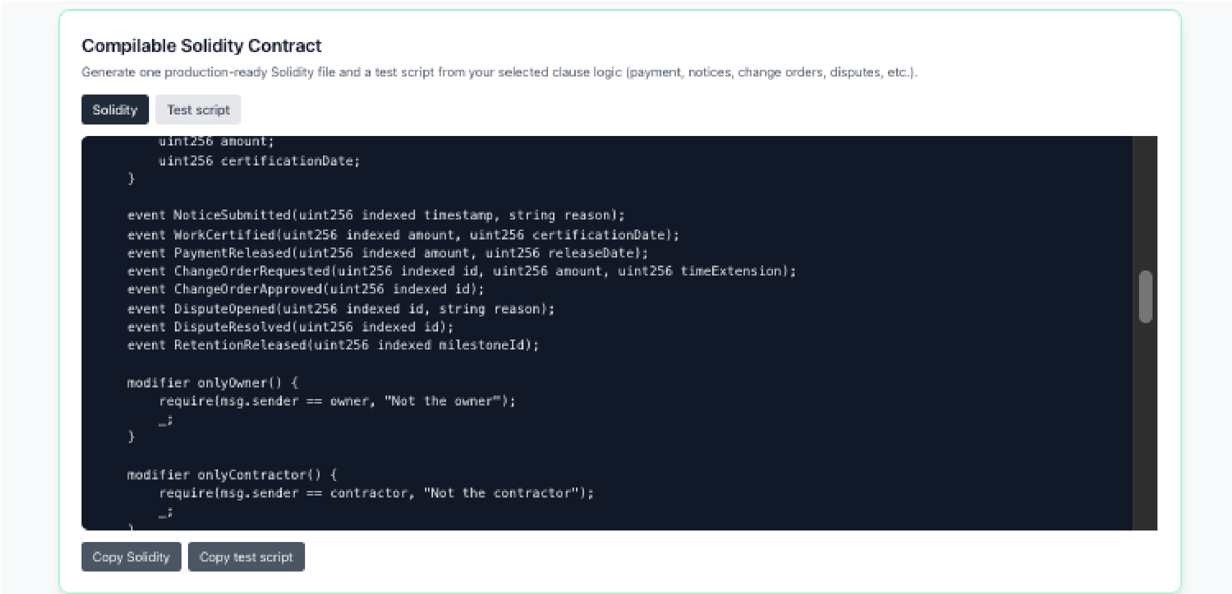


Figure 9: Smart Contract logic synthesis Tab.

To further illustrate end-to-end operation, a representative clause from a commissioning contract specified that reimbursable expenses such as travel and reproduction were included within a lump-sum fee and not separately reimbursable. While this provision addressed cost allocation, the system identified partial misalignment with standard payment practice because it lacked milestone-based payment triggers and delayed-payment interest protections.

Table 9: Example of Clause-Level Analysis and Remediation.

Component	**Description**
Clause Category	Payment
Issue Type	Missing provision
Description	No milestone payments; no interest on delayed payments
Severity	0.75
Affected KPIs	CFA (primary), SDI (secondary)
Importance	High
Suggested Remedy	Add milestone-based payments and late-payment interest
Output	Smart contract logic for milestone certification and payment release

The framework classified this issue as a high-severity missing-type discrepancy primarily affecting cash-flow adequacy. It then generated targeted remediation logic introducing milestone certification events, payment deadlines, and late-payment interest conditions. This example demonstrates how clause semantics, standards alignment, and KPI reasoning can be translated into operational governance logic. Table 9 summarizes the clause-level diagnosis and generated remedy.

To assess whether generated logic could be operationalized beyond static template generation, a lightweight execution-level validation was performed on three representative template families. Across six simulated cases, including both triggering and non-triggering conditions, all generated templates behaved as expected, yielding a 100% pass rate. This provides preliminary evidence that the synthesized logic is not only structurally coherent but also executable under simple state conditions. Detailed results are reported in Appendix E.

These results indicate that the proposed framework functions as a decision-support and governance automation tool rather than a deterministic predictor of project outcomes. Its primary value lies in identifying where automation is most worthwhile, structuring enforceable workflows, and prioritizing corrective interventions in high-impact contractual provisions.

5. DISCUSSION

The clause importance rankings reveal a highly uneven distribution of contractual influence across project performance indicators. A relatively small subset of provisions accounts for a disproportionate share of modeled relevance to cost overruns, schedule delays, cash-flow adequacy, and dispute frequency. This concentration effect is consistent across contract types and reinforces longstanding industry observations that payment administration, approvals, notices, and change management mechanisms frequently shape downstream project performance. The present study extends prior qualitative understanding by providing a quantitative, clause-level attribution framework rather than relying solely on aggregate contract categories or narrative assessments. Importantly, this concentration pattern remained stable across alternative KPI weighting assumptions and classification stress scenarios, suggesting it reflects a structural characteristic of how governance significance is distributed within construction contracts rather than an artifact of any specific modeling choice.

The identification of many clauses with negligible or limited marginal contribution also highlights a practical limitation of traditional contract review processes, which often allocate similar attention to all provisions regardless of their governance significance. By contrast, the proposed framework enables prioritization of review, negotiation, and automation efforts toward clauses with the greatest modeled performance relevance. This selective orientation is particularly relevant given the scale and heterogeneity of public-sector construction contracts, where exhaustive clause-by-clause review is both resource-intensive and prone to oversight.

An important methodological finding concerns the relative performance of domain-calibrated classification versus zero-shot LLM prompting. Inter-annotator analysis revealed that the original ten-category taxonomy produced moderate agreement (Cohen's $\kappa = 0.52$), with disagreement concentrated in procedurally entangled categories where provisions function as prerequisites or consequences of adjacent governance mechanisms. Taxonomy revision to eight categories, motivated by this annotation evidence, improved macro-averaged F1 from 0.550 to 0.572 on the full corpus and to 0.718 on a balanced validation set. A chain-of-thought LLM prompt evaluated on the same balanced set achieved only 0.483, substantially underperforming the domain-calibrated classifier. The LLM systematically over-predicted general conditions when uncertain, defaulting to the broadest available category in the absence of domain-specific training. This pattern is consistent with findings from a companion retrieval study on the same corpus, where structured approaches outperformed naive semantic baselines (Moayyed & Anumba, 2026), and confirms that construction contract analysis requires domain-specific calibration rather than general LLM capability. This finding has implications beyond this study: it suggests that the deployment of LLM-based contract tools in practice should not assume that general prompting strategies are sufficient without domain adaptation.

The standards alignment analysis further indicates that contractual risk in this corpus frequently arises through omission, weakening, or fragmentation of governance safeguards rather than through direct contradiction of standard forms; this pattern is an observation from the analyzed contracts rather than a demonstrated characteristic of construction contracts generally. Initial clause-level results showed a high prevalence of missing-type findings, particularly in payment and administrative provisions. However, the subsequent contract-context audit demonstrated that many such findings may reflect distributed drafting structures rather than definitive contract-level absence. This distinction matters practically. It suggests that standards alignment should be interpreted as a diagnostic signal requiring contextual review rather than a binary compliance judgment, and that automated discrepancy detection tools should incorporate contract-level context rather than operating at the clause level alone.

The prominence of cash-flow adequacy as the dominant KPI affected across all three discrepancy types warrants a qualified interpretation. Part of this pattern reflects the corpus and modeling structure directly: payment clauses constitute the largest clause category in this corpus and carry the highest CFA linkage values by design in the rule-based KPI mapping, so discrepancies concentrated in payment-related provisions are systematically attributed to CFA regardless of any independent substantive signal. Beyond this structural feature, construction contracts are often evaluated primarily through cost and schedule lenses, yet the results suggest that payment timing, certification mechanisms, and liquidity protections function as foundational control variables influencing multiple downstream outcomes, including schedule continuity and dispute exposure. This observation is broadly consistent with system dynamics modeling of construction governance, which identifies payment friction as the primary transmission channel through which macroeconomic volatility amplifies cost and schedule escalation (Moayyed et al., 2026). Because both analyses draw on the same institutional corpus and the same research program, this consistency should be read as coherence within a shared research program rather than corroboration from methodologically independent sources; broader empirical validation across owner types, and ideally across research groups, remains necessary before generalizing this pattern.

The projected benefits of remediation and selective automation should be interpreted cautiously. The reported values are normalized scenario-based indicators derived from modeled linkage relationships rather than observed post-implementation project outcomes. Their primary purpose is comparative prioritization: identifying where corrective intervention or automation is likely to be most valuable under explicit assumptions. Even under more conservative remediation effectiveness scenarios, ranking stability remained high, indicating that the framework is more informative for clause selection than for predicting exact magnitudes of benefit.

The smart contract synthesis component should be understood as structured workflow compilation rather than autonomous legal replacement. The strongest use case lies in operationalizing recurring administrative controls such as payment deadlines, notice periods, milestone certifications, retainage release, and escalation triggers. The substantial improvement achieved through constrained template-based synthesis suggests that governance automation in construction may be more practical when focused on bounded operational functions rather than unrestricted text-to-code generation. This is consistent with recent evidence from legal NLP that domain-specific and task-adapted models can outperform larger general-purpose LLMs on contract-understanding tasks despite substantially fewer parameters (Singh et al., 2025), suggesting that bounded, domain-specific automation may be a more reliable path than unconstrained generative approaches for high-stakes contractual workflows.

From a theoretical perspective, this study reframes construction contracts as performance-sensitive governance systems rather than static legal artifacts. By integrating clause semantics, KPI relevance, standards alignment, and executable logic into a unified proof-of-concept framework, the research connects contract theory, project controls, and computational governance within a single analytical structure. Methodologically, it demonstrates how retrieval-augmented language models can be embedded within structured reasoning pipelines to support explainable and auditable contract analytics, while also establishing that the boundaries of current LLM capability require domain-specific adaptation before reliable deployment in construction contract settings.

For practitioners, the findings support a shift from exhaustive document review toward impact-driven governance management. Rather than treating all clauses equally, owners, contractors, and consultants may allocate negotiation effort, compliance monitoring, and automation resources toward a relatively small set of high-leverage provisions. This selective approach may deliver substantial practical value without requiring wholesale contract digitization or full replacement of existing administration processes. The finding that governance risk enters primarily through omission and weakening rather than contradiction also has direct implications for contract drafting guidance: systematic comparison against reference standards during drafting may prevent the most consequential governance gaps before execution.

Several limitations should be acknowledged. First, the empirical evaluation is based on five executed contracts from a single public owner, which constrains generalizability across owner types, jurisdictions, and procurement cultures. Corpus analysis using a companion retrieval study on the same contracts revealed that 81.5% of clauses belong to a broad general conditions category, reflecting the drafting practices of this specific institutional context rather than the construction industry broadly. Transferability beyond this context is not established: the standards-alignment module is built around AIA A201, and applying the framework to contracts governed by other reference forms (e.g., FIDIC internationally, ConsensusDocs, or public-sector model conditions outside the U.S.) would require rebuilding the standards repository and re-validating the discrepancy taxonomy against the new reference form. Private-sector procurement, which often involves different risk-allocation and payment norms than the public-sector corpus analyzed here, may likewise exhibit different clause-importance and discrepancy patterns. Second, KPI linkage values and importance weights are design assumptions grounded in construction management literature rather than empirically calibrated parameters derived from observed project outcome data. The KPI weight vector should therefore be interpreted as a structured prioritization tool rather than a predictive model of actual performance effects. Third, classification performance is bounded by the inherent semantic ambiguity of construction contract language, reflected in moderate inter-annotator agreement ($\kappa = 0.52$), which establishes a practical ceiling for any automated classifier on this taxonomy. Fourth, zero-shot LLM prompting underperformed the domain-calibrated classifier substantially, indicating that current generative approaches require domain-specific fine-tuning before they can serve as reliable classification tools in this setting. Fifth, smart contract synthesis was evaluated through internal rubric assessment and simulation rather than deployment in live contract administration environments. Sixth, and more fundamentally, the validation strategy applied throughout this study, spanning the clause taxonomy, KPI weighting scheme, severity scoring, and template rules, demonstrates that the framework operates consistently with its own internal assumptions rather than establishing external validity. The evaluation confirms internal coherence, for example, stable clause rankings under sensitivity analysis and consistent aggregation behavior across scenarios, but it does not establish whether the framework improves real contract administration outcomes or predicts project performance more effectively than existing approaches. Assessing external validity would require prospective deployment studies comparing framework-guided decisions against baseline practice, which is beyond the scope of this proof-of-concept and is identified as a priority for future work.

Beyond these limitations, computational scalability warrants consideration for deployment on larger contract repositories. Clause extraction and embedding-based retrieval scale approximately linearly with corpus size, while the discrepancy-detection and importance-scoring stages that invoke language-model inference incur per-clause processing cost and latency; for repositories spanning thousands of contracts, this implies a practical trade-off between batch processing (favoring throughput) and interactive review (favoring latency). It also suggests that selective, importance-ranked processing, including prioritizing high-leverage clauses identified early in the pipeline, rather than exhaustive first-pass analysis of every clause, may be the more practical deployment pattern at scale.

Future research should address these limitations through several directions. Expanding the corpus across multiple owner types, jurisdictions, and contract families is the most critical priority for establishing generalizability, and a companion validation study with an extended corpus is currently underway. Expert elicitation of KPI weights through structured practitioner surveys would provide independent empirical grounding for the prioritization framework. Domain-specific fine-tuning of legal language models on construction contract corpora, using pre-training resources such as the CUAD dataset combined with construction-specific annotation, represents the most direct path to improving classification performance beyond the current ceiling. Integration with live project data streams, including payment records, RFIs, and schedule updates, would enable real-time governance monitoring rather than static contract analysis. Additional work may explore multi-party dispute resolution logic, cross-contract dependency management, and interoperability with BIM-based digital twins, building on recent frameworks integrating BIM and blockchain-based smart contracts for construction payment administration (Sonmez et al., 2022). By establishing a proof-of-concept foundation and identifying the specific empirical and methodological requirements for scaled validation, this study positions the proposed framework as a starting point for a broader research program in performance-driven construction contract governance.

6. CONCLUSIONS

This study proposed and preliminarily evaluated a Retrieval-Augmented Generation-based framework for clause-level contract intelligence and structured smart contract synthesis in construction projects. The framework integrates clause extraction, semantic classification, KPI-based importance modeling, standards alignment, discrepancy diagnostics, and constrained logic synthesis within a unified proof-of-concept pipeline. In contrast to prior studies that examine contract analytics, blockchain automation, or document intelligence in isolation, the framework connects three previously fragmented layers: contract language, project performance priorities, and executable governance workflows.

The study's four preliminary findings describe governance risk and remediation opportunity as both concentrated rather than evenly distributed across contract text. A small subset of payment and approval provisions carries most of the modeled performance relevance, and where those high-leverage provisions deviate from reference standards, the deviation is typically omission or a weakened obligation rather than outright contradiction, a pattern that points toward standards-based review, rather than adversarial dispute resolution, as the more appropriate corrective mechanism. On the automation side, the same underlying principle of constraint proved decisive in two independent tests: template-constrained smart contract synthesis outperformed unconstrained generation, and a domain-calibrated classifier outperformed zero-shot LLM prompting, each by a wide margin. These results suggest that within this five-contract, single-owner corpus, performance-driven contract governance benefits less from broader generative capability than from narrowing that capability to the specific structural regularities, including payment concentration, omission-based risk, and bounded template logic, that the corpus reveals.

Beyond these four findings, this study contributes conceptually and methodologically to construction contract informatics. Conceptually, it introduces a clause-to-KPI reasoning architecture that connects individual contract provisions to project performance dimensions, moving analysis from document-centric classification toward performance-driven governance reasoning. Methodologically, it demonstrates that retrieval-augmented reasoning, KPI-weighted importance scoring, and constrained template synthesis can be integrated into a single auditable pipeline, and it provides preliminary empirical evidence (within the scope of a single public-owner corpus) that governance risk in construction contracts concentrates in omission and weakening of provisions rather than direct contradiction of reference standards.

For practice, these findings support directing negotiation effort, compliance monitoring, and automation investment toward the relatively small set of high-leverage payment and approval provisions identified here, rather than reviewing all clauses with uniform scrutiny. Because governance risk arises primarily through omission and weakening rather than outright contradiction, systematic comparison against reference standards during drafting, rather than only during dispute resolution, may help contract managers, owners, and consultants prevent the most consequential governance gaps before execution.

These findings should be interpreted within the scope of a proof-of-concept evaluation. The empirical basis is five contracts from a single public owner, KPI improvement estimates are modeled scenario indicators rather than observed project outcomes, and deployment within live enterprise environments was outside the scope of this study. The classification ceiling is further bounded by the inherent semantic ambiguity of construction contract language, reflected in moderate inter-annotator agreement.

This study represents Stage 1 of a two-phase research program. A companion validation study is currently underway to evaluate the framework across an expanded multi-owner corpus, compare classification performance against domain-fine-tuned language models, and validate KPI weights through expert elicitation. Additional directions include integration with live project data streams, multi-party dispute resolution logic, and interoperability with BIM-enabled digital twins. By establishing the architectural foundation, identifying the empirical requirements for broader validation, and producing preliminary evidence on governance concentration patterns and domain adaptation requirements, this study provides a structured starting point for performance-driven contract intelligence in construction.

REFERENCES

Chalkidis, I., & Kampas, D. (2019). Deep learning in law: early adaptation and legal word embeddings trained on large corpora. *Artificial Intelligence and Law*, *27*(2), 171–198. https://doi.org/10.1007/s10506-018-9238-9

Chan, A. P. C., Scott, D., & Chan, A. P. L. (2004). Factors affecting the success of a construction project. *Journal of Construction Engineering and Management*, *130*(1), 153–155.

Cheung, S. O., & Pang, H. Y. (2013). Anatomy of construction disputes. *Journal of Construction Engineering and Management*, *139*(2), 15–23.

Elkhayat, Y., & Marzouk, M. (2022). Selecting feasible standard form of construction contracts using text analysis. *Advanced Engineering Informatics*, *52*, 101569. https://doi.org/10.1016/j.aei.2022.101569

Flyvbjerg, B., Holm, M. K. S., & Buhl, S. L. (2004). What causes cost overrun in transport infrastructure projects? *Transport Reviews*, *24*(1), 3–18.

Hamledari, H., & Fischer, M. (2021). Measuring the impact of blockchain and smart contracts on construction supply chain visibility. *Advanced Engineering Informatics*, *50*, 101444. https://doi.org/10.1016/j.aei.2021.101444

Holzenberger, N., Van Durme, B., & Singh, S. (2021). Efficient rule extraction for legal reasoning. *Proceedings of ACL*.

Izacard, G., & Grave, E. (2021). Leveraging passage retrieval with generative models for open domain question answering. *Proceedings of the European Chapter of the ACL*.

Lee, J., Lee, S., & Kim, B. (2016). Automated checking of construction contract compliance using rule-based reasoning. *Automation in Construction*, *65*, 1–14.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems*, *33*, 9459–9474.

Li, J., Greenwood, D., & Kassem, M. (2019). Blockchain in the built environment and construction industry: A systematic review, conceptual models and practical use cases. *Automation in Construction*, *102*, 288–307.

Love, P. E. D., Edwards, D. J., Irani, Z., & Sharif, A. (2016). Participatory action research approach to public sector procurement selection. *International Journal of Project Management*, *34*(4), 512–528.

Mitkus, S., & Mitkus, T. (2014). Causes of conflicts in a construction industry: A communicational approach. *Procedia Social and Behavioral Sciences*, *110*, 777–786.

Moayyed, E., & Anumba, C. J. (2024). Utilizing Large Language Models for Data Extraction from Construction Contracts: A Pathway to Generating Blockchain-Based Smart Contracts. *Computing in Civil Engineering*, 363–372.

Moayyed, E., & Anumba, C. J. (2026, May 20). Evaluating Retrieval Augmented Generation for Clause Level Information Retrieval in Construction Contracts. *International Conference on Computing in Civil Engineering (i3CE 2026)*, Songdo, Incheon, South Korea. https://doi.org/10.5281/zenodo.20313645

Moayyed, E., Anumba, C. J., & Castelblanco, G. (2026). Adaptive Contract Governance: Stabilizing Construction Project Performance against Macroeconomic Volatility using Smart Contract Automation. *Journal of Management in Engineering*. https://doi.org/10.1061/JMENEA.MEENG-7491

Moayyed, E., Anumba, C. J., & Morteza, A. (2025). Systematic analysis of large language models for automating document-to-smart contract transformation. *Automation in Construction*, *175*, 106209.

Nasser, M., Hamdy, A., & Marzouk, M. (2026). Analyzing construction contract selection factors: A semantic analysis using NLP. *Advanced Engineering Informatics*, *69*, 103810. https://doi.org/10.1016/j.aei.2025.103810

Singh, A., Karaca, H. S., Joshi, A., Paik, H., & Jiang, J. (2025). LLMs for law: Evaluating legal-specific LLMs on contract understanding. *arXiv preprint arXiv:2508.07849*. https://doi.org/10.48550/arXiv.2508.07849

Sonmez, R., Ahmadisheykhsarmast, S., & Güngör, A. A. (2022). BIM integrated smart contract for construction project progress payment administration. *Automation in Construction*, *139*, 104294.

Turk, Ž., & Klinc, R. (2017). Potentials of blockchain technology for construction management. *Procedia Engineering*, *196*, 638–645.

Un, B., Erdis, E., Aydinli, S., Genc, O., & Alboga, O. (2025). Forecasting the outcomes of construction contract disputes using machine learning techniques. *Engineering, Construction and Architectural Management*, *32*(10), 6421–6444.

Wang, J., Wu, P., Wang, X., & Shou, W. (2020). The outlook of blockchain technology for construction engineering management. *Frontiers of Engineering Management*, *7*(1), 67–75.

Wu, C., Ding, W., Jin, Q., Jiang, J., Jiang, R., Xiao, Q., Liao, L., & Li, X. (2025). Retrieval augmented generation-driven information retrieval and question answering in construction management. *Advanced Engineering Informatics*, *65*, 103158. https://doi.org/10.1016/j.aei.2025.103158

Yiu, T. W., & Cheung, S. O. (2006). A catastrophe model of construction conflict behavior. *Building and Environment*, *41*(4), 438–447.

Zhang, J., & El-Diraby, T. (2012). A semantic ontology-based framework for knowledge management in construction. *Journal of Computing in Civil Engineering*, *26*(4), 541–555.

Zhang, L., & Ning, Y. (2026). Improving construction contract question answering through embedding optimization and semantic chunking in large language models. *Advanced Engineering Informatics*, *69*, 104027. https://doi.org/10.1016/j.aei.2025.104027

Zhao, X., Hwang, B. G., & Yu, G. (2022). Contractual governance and performance in complex construction projects. *International Journal of Project Management*, *40*(3), 230–244.

Zhou, S., Xie, S., Yi, W., & Wang, W. (2026). Construction case-relevant article-level law identification using fine-tuned large language models: A study in China's construction industry. *Advanced Engineering Informatics*, *70*, 104144. https://doi.org/10.1016/j.aei.2025.104144

APPENDIX A: CLAUSE TAXONOMY AND ANNOTATION PROTOCOL

A.1 Clause taxonomy

To support clause-level analysis, a predefined taxonomy of contractual categories was developed based on common structures in construction contracts and prior literature on contract administration. The taxonomy consists of ten categories:

- **Payment**: provisions governing invoicing, payment timing, retainage, and financial flows
- **Schedule**: provisions related to timelines, milestones, delays, and extensions
- **Change Orders**: provisions governing scope changes, variation procedures, and approvals
- **Risk Allocation**: clauses assigning responsibility for risks, liabilities, and indemnification
- **Submittals and Approvals**: provisions related to document submission, review, and approval workflows
- **Disputes and Claims**: provisions governing claims, dispute resolution, and escalation mechanisms
- **General Conditions**: general administrative and procedural clauses not specific to a single function
- **Other**: clauses that do not clearly fall into the above categories or have negligible operational impact

Each clause is assigned to a single category for consistency with downstream modeling. Following inter-annotator analysis which revealed that agreement was lowest in procedurally entangled categories, the taxonomy was revised from ten to eight categories. Termination provisions were merged into Risk Allocation, reflecting their function as a risk consequence rather than a standalone governance mechanism. Notice provisions were merged into Disputes and Claims, reflecting their role as procedural prerequisites to claims.

A.2 Annotation protocol

A subset of 120 clauses was randomly sampled from the full corpus for manual validation. Each clause was independently labeled according to the predefined taxonomy by annotators familiar with construction contract administration.

The annotation process followed three key principles:

1. **Primary functional role**

   Each clause was assigned to the category that best reflects its dominant contractual function. When clauses contained multiple elements, the classification was based on the primary operational purpose rather than secondary references.

2. **Single-label assignment**

   Although many contractual provisions are multifunctional, a single-label approach was adopted to ensure consistency with the classification model and to enable quantitative evaluation using precision, recall, and F1 metrics.

3. **Context-aware interpretation**

   Clauses were interpreted within their immediate textual context, including headings and surrounding provisions, to resolve ambiguous phrasing or cross-references.

A.3 Handling ambiguous clauses

Certain clause categories, such as general conditions, submittals and approvals, and other, exhibit inherently higher semantic overlap. For such clauses, annotators applied the following decision rules:

- If a clause describes a specific operational process (e.g., submittal review workflow), it is assigned to that category rather than general conditions.

- If a clause contains multiple functions, the category corresponding to the primary enforceable obligation is selected.
- If no dominant functional role can be identified, the clause is assigned to other.

These rules were introduced to reduce inconsistency while acknowledging that strict categorical separation is not always achievable in legal text.

APPENDIX B: CLASSIFICATION VALIDATION AND BASELINE COMPARISON

B.1 Validation dataset and evaluation protocol

The validation setup is described in Section 4.2. Classification performance was assessed on a manually labeled subset of 120 clauses using precision, recall, F1, and the macro-averaged F1 score; the category taxonomy is provided in Appendix A.

B.2 Proposed model performance

Table B1 reports classification performance across all ten clause categories using the proposed hybrid classification approach.

Table B1: Classification performance by category.

Category	Precision	Recall	F1
Schedule	0.732	0.612	0.667
Change Orders	0.714	0.588	0.645
Payment	0.619	0.639	0.629
Risk Allocation	0.719	0.561	0.630
Submittals & Approvals	0.656	0.553	0.600
Other	0.753	0.416	0.536
General Conditions	0.627	0.396	0.486
Disputes & Claims	0.375	0.500	0.429

The macro-averaged F1 score across all categories is 0.550, indicating moderate overall classification performance across heterogeneous clause types.

Performance is strongest in structurally explicit categories such as *schedule*, *change orders*, and *payment*, and weaker in categories such as *notices*, *disputes and claims*, and *termination*. These lower-performing categories exhibit higher semantic overlap and multifunctional structure, making strict classification inherently challenging.

B.3 Inter-annotator agreement

To assess labeling reliability, the validation subset was independently annotated by two reviewers. Agreement metrics were computed as follows:

- **Percent agreement**: 69.17%
- **Cohen's κ**: 0.52 (moderate agreement)

Disagreement was concentrated in boundary-prone categories such as *submittals and approvals*, *general conditions*, and *other*, indicating that classification ambiguity is driven by inherent characteristics of contractual language rather than solely by model limitations.

B.4 Baseline comparison

To evaluate the effectiveness of the proposed classification approach, two lightweight baseline methods were implemented and tested on the same validation subset:

1. **Rule-based baseline**: keyword matching using predefined lexical cues (e.g., "payment," "schedule," "notice")
2. **Embedding-only baseline**: TF-IDF vector similarity between clause text and category descriptions

Table B2 presents the comparative performance of these methods.

Table B2: Baseline comparison (macro-averaged metrics).

Method	Precision	Recall	F1
Proposed hybrid model	0.55	0.55	0.55
LLM prompt v2 (chain-of-thought)	0.48	0.49	0.483
Embedding-only baseline	0.198	0.163	0.147
Rule-based baseline	0.137	0.175	0.136

The LLM prompt v2 was evaluated on the same balanced validation set of 141 clauses; all other methods were evaluated on the original 120-clause set. The proposed hybrid model significantly outperforms both baselines, achieving over 300% relative improvement in F1 score compared to simpler approaches. This demonstrates that the integration of retrieval-augmented reasoning and structured taxonomy constraints provides substantial gains over purely lexical or similarity-based classification methods.

B.5 Classification robustness analysis

Because downstream components depend on classification outputs, additional robustness testing was conducted to assess the sensitivity of results to classification uncertainty in lower-performing categories.

Two stress-test scenarios were evaluated:

1. **Exclusion test**: clauses belonging to low-F1 categories (*notices*, *disputes and claims*, *termination*, *general conditions*) were removed from the analysis
2. **Confidence-weighted test**: clause contributions were weighted by category-level F1 scores to reflect classification confidence

The results show that:

- Core qualitative patterns remain stable across both tests
- Payment-related clauses continue to dominate top-ranked clauses
- Cash-flow adequacy (CFA) remains the dominant KPI pattern
- Ranking similarity remains high (Spearman correlation up to 0.98)

Quantitatively, stress conditions reduce absolute magnitudes of projected improvements by approximately 8–10%, while preserving overall ranking structure.

APPENDIX C: KPI LINKAGE AND IMPACT MODELING

C.1 Rule-based KPI mapping

The rule-based component assigns baseline KPI influence patterns based on clause category. Table C1 summarizes the primary mappings.

Table C1: Rule-based clause category to KPI mapping.

Category	COI	SDI	CFA	DF
Payment	High	Medium	High	Low
Schedule	Medium	High	Low	Low
Change Orders	High	Medium	Medium	Medium
Risk Allocation	High	Low	Low	Medium
Submittals & Approvals	Medium	High	Medium	Medium
Disputes & Claims	Medium	Medium	Low	High
Termination	Medium	Low	Medium	High
Notices	Low	Medium	Low	High
General Conditions	Low	Low	Low	Medium
Other	Low	Low	Low	Low

These mappings provide an initial allocation of influence across KPIs based on established construction management principles.

C.2 Example clause–KPI mappings

To illustrate the linkage process, Table C2 presents representative examples.

Table C2: Example clause–KPI linkage mappings.

Clause Type	Description	COI	SDI	CFA	DF
Payment	Monthly invoice within 30 days	0.30	0.20	0.80	0.10
Schedule	Delay notification requirement	0.10	0.70	0.10	0.40
Disputes	Formal dispute escalation clause	0.20	0.20	0.10	0.80
Submittals	Approval workflow for shop drawings	0.20	0.60	0.30	0.30

These examples demonstrate how clause semantics influence multiple KPIs simultaneously and highlight the multidimensional nature of contractual impact.

C.3 KPI impact estimation and normalization

Clause-level discrepancies are translated into modeled KPI improvements using:

$$\Delta \mathrm{KPI}_{c,k} = l_{c,k} \cdot s_c \cdot r_c$$

where:

- s_c is discrepancy severity
- r_c is remediation effectiveness

Direct summation of clause-level improvements yields an unbounded cumulative impact index. To ensure interpretability, a normalized project-level score is defined as:

$$NS = \frac{\sum_{c \in C} I(c)}{\sum_{c \in C} I_c^{\max}}$$

where $I_c^{\max}$ represents the maximum attainable contribution for clause c under full severity and full remediation.

This normalized score represents the fraction of achievable improvement under the modeled linkage structure.

C.4 Sensitivity to remediation effectiveness

To evaluate robustness, the remediation coefficient r_c was varied across three levels:

- r_c = 1.00 (baseline)
- r_c = 0.75
- r_c = 0.50

Results show that:

- Absolute impact values scale proportionally with r_c
- Clause ranking remains highly stable (Spearman correlation > 0.99)
- Top-ranked clauses remain unchanged across scenarios

This indicates that the model captures structurally important clauses rather than artifacts of parameter selection.

APPENDIX D: STANDARDS ALIGNMENT AND MISSING PROVISION AUDIT

D.1 Contract-context audit methodology

For each discrepancy initially classified as missing, a second-pass analysis was performed to evaluate whether the required obligation was addressed elsewhere in the same contract.

This process consists of three steps:

Step 1: Obligation inference

An obligation concept is inferred from:

- discrepancy description
- associated standard reference
- suggested remedy
- clause context

Examples of inferred concepts include:

- payment timing requirement
- notice deadline
- milestone-based payment
- dispute escalation procedure

Step 2: Cross-clause search

The framework searches other clauses within the same contract using:

- keyword matching (e.g., “payment,” “invoice,” “notice”)
- token overlap similarity
- contextual relevance (section proximity and semantic similarity)

Step 3: Evidence-based relabeling

Based on the strength of evidence, each discrepancy is reassigned to one of three categories:

- **Confirmed missing**: no meaningful evidence of the obligation elsewhere
- **Possibly relocated**: partial or indirect evidence of the obligation in other clauses
- **Uncertain**: weak or ambiguous evidence insufficient for confident classification

D.2 Threshold configurations

Two threshold configurations were evaluated:

- **Strict setting**: requires strong and specific evidence to classify as “possibly relocated”
- **Moderate setting**: allows broader lexical and contextual matches

These configurations provide a sensitivity analysis for interpreting missing provisions.

D.3 Audit results

Table D1 summarizes the results of the contract-context audit.

Table D1: Missing provision audit results.

Setting	Confirmed Missing	Possibly Relocated	Uncertain
Strict	19	98	228
Moderate	19	161	165

Under the strict configuration, only a small subset of initially missing provisions remain confirmed as true omissions, while a substantial portion is either potentially addressed elsewhere or remains ambiguous.

D.4 Category-level observations

The audit results reveal several important patterns:

- Payment-related clauses dominate missing findings, reflecting the distributed nature of financial provisions across contracts
- Notice-related obligations are frequently relocated, often embedded within administrative or procedural clauses
- General conditions clauses exhibit lower ambiguity, as they tend to consolidate broader governance provisions

These findings indicate that clause-level absence is often a structural characteristic of contract organization rather than a direct indicator of governance deficiency.

D.5 Representative examples

Table D2 presents representative audit outcomes.

Table D2: Example missing provision audit cases.

Clause ID	Inferred Concept	Audit Result	Evidence
Clause A	Notice deadline	Possibly relocated	Notice requirements found in multiple administrative sections
Clause B	Payment obligation	Possibly relocated	Payment terms distributed across invoice and audit clauses
Clause C	Payment timing	Uncertain	Partial lexical overlap but no clear obligation
Clause D	Termination condition	Confirmed missing	No evidence found across contract

These examples illustrate how contractual obligations may be fragmented across multiple sections and highlight the importance of context-aware interpretation.

APPENDIX E: CONSTRAINED SMART CONTRACT SYNTHESIS AND VALIDATION

E.1 Template-based synthesis architecture

The constrained synthesis process consists of three sequential components:

(1) Remedy Classification

Each discrepancy and associated corrective recommendation is assigned to a predefined logic family, such as payment enforcement, notice timing, milestone release, change approval, or dispute escalation.

(2) Slot Extraction

Relevant contractual parameters are extracted from clause text and remedy descriptions. Typical parameters include:

- time thresholds (e.g., 14 days, 30 days)
- monetary terms (e.g., retainage, interest rate, payment amount)
- responsible actors (e.g., owner, contractor, architect)
- triggering conditions (e.g., invoice approval, milestone completion)
- escalation conditions (e.g., failure to respond within deadline)

(3) Template Instantiation

The extracted parameters are inserted into structured logic templates representing enforceable contractual rules. This produces consistent machine-readable outputs while preserving clause-specific meaning.

E.2 Template library

A domain-specific library of recurring contractual mechanisms was developed. Representative templates are summarized in Table E1.

Table E1: Representative smart contract templates.

Template Type	Description	Example Parameters
Payment Due	Enforces payment within specified period	due_days, payer, payee
Late Payment Interest	Applies interest to overdue balances	interest_rate, due_days
Milestone Payment	Releases payment after milestone verification	milestone_id, amount
Notice Deadline	Requires notice within defined period	notice_days, triggering_event
Change Order Approval	Governs approval of scope changes	authority, time_limit
Dispute Trigger	Initiates escalation procedure	dispute_condition

These templates encode common governance mechanisms repeatedly observed in construction contracts and provide a structured bridge between legal text and executable logic.

E.3 Example clause-to-logic transformation

Table E2 illustrates the translation of a payment clause into structured smart contract logic.

Table E2: Example clause-to-logic transformation.

Step	Description
Clause	“Payment shall be made within 30 days of invoice submission.”

Step	Description
Detected Issue	Missing enforcement of payment timing
Suggested Remedy	Add timely payment trigger and overdue penalty
Selected Templates	Payment Due + Late Payment Interest
Extracted Slots	due_days = 30; interest_rate = contract-defined
Generated Logic	Release payment within 30 days; apply interest if unpaid after due date

This example demonstrates how contractual semantics can be systematically transformed into enforceable rule structures.

E.4 Validation methodology

Two complementary validation procedures were conducted.

(1) Rubric-Based Quality Evaluation

A representative sample of 10 clause–remedy pairs was evaluated using four criteria:

1. **Semantic consistency** – alignment between generated logic and original clause meaning
2. **Remedy alignment** – extent to which generated logic reflects the recommended correction
3. **Structural validity** – coherence and completeness of rule structure
4. **Condition–action completeness** – presence of explicit triggers and corresponding actions

Scores were aggregated to compare unconstrained generation with template-based synthesis.

(2) Lightweight Execution-Level Validation

To assess whether generated logic could be operationalized beyond static text generation, representative templates were executed under simulated triggering and non-triggering scenarios. Three template families were tested:

- Payment Due
- Late Payment Interest
- Milestone Payment

Each template was evaluated using one triggering case and one non-triggering case.

E.5 Validation results

Table E3 compares synthesis quality before and after the introduction of the constrained template architecture.

Table E3: Smart contract synthesis quality comparison.

Metric	Unconstrained Generation	Template-Based Synthesis
Semantic consistency	0.00	0.60
Remedy alignment	0.00	1.00
Structural validity	0.00	1.00
Condition–action completeness	0.80	1.00
Overall score (0–4)	0.80	3.60

The constrained approach substantially improved all quality dimensions, increasing the overall score from 0.80 to 3.60 out of 4.00.

E.6 Execution-level validation

Table E4 summarizes simulated execution outcomes for representative templates.

Table E4: Execution-level validation of generated templates.

Template Type	Scenario	Expected Outcome	Result
Payment Due	Overdue unpaid invoice	trigger_payment = True	Pass
Payment Due	Invoice not yet due	trigger_payment = False	Pass
Late Payment Interest	Unpaid after due date	apply_interest = True	Pass
Late Payment Interest	Paid before due date	apply_interest = False	Pass
Milestone Payment	Milestone approved	release_payment = True	Pass
Milestone Payment	Milestone incomplete	release_payment = False	Pass

All six scenarios passed, yielding a 100% success rate for the tested template set. These results provide preliminary evidence that generated rules are not only structurally coherent but also executable under simple state conditions.

E.7 Limitations

Several limitations remain.

- Template coverage is currently limited to common contractual mechanisms. Highly specialized clauses may require new templates.
- Slot extraction relies partly on heuristic parsing and may require refinement for complex legal language.
- Validation used a small internal sample and a lightweight simulation rather than blockchain deployment, adversarial testing, or live project execution.
- The synthesis module should therefore be interpreted as a structured translation mechanism rather than a production-ready autonomous contracting system.