Automation in Construction 191 (2026) 107165

Contents lists available at ScienceDirect

Automation in Construction

journal homepage: www.elsevier.com/locate/autcon

Review

Automated compliance checking in AEC in the era of AI and LLMs: A review (2022–2025)

Youssef Senousy [a,*], Franco Cheung [a], Thomas Beach [b], Ogerta Elezaj [a], Edlira Vakaj [a]

[a] School of Architecture, Built Environment, Computing and Engineering, Birmingham City University, Birmingham, UK
[b] School of Engineering, Cardiff University, Cardiff, UK

ARTICLE INFO

Dataset link: https://github.com/ysenousy/ACC-N-ACE

Keywords:
Artificial Intelligence (AI)
AutomateD Compliance Checking (ACC)
Building Information Modelling (BIM)
Explainability
Human-in-the-loop (HITL)
Large Language Models (LLMs)

ABSTRACT

Since 2022, advances in Artificial Intelligence (AI) and Large Language Models (LLMs) have reshaped Automated Compliance Checking (ACC) in AEC. This review applies an AI-driven PRISMA workflow combining LLM-assisted discovery and screening with transparent provenance and audit trails. Studies are mapped to five ACC pipeline stages: rule interpretation, model preparation, rule execution, reporting, and decision support. Iterative coding identifies ten cross-cutting themes used as analytical lenses: Multimodal Information Extraction, Formalisation of Regulatory Text, Semantic Alignment with BIM/IFC, Integration of Ontologies and Knowledge Graphs, Rule Representation and Reasoning, Model-Driven Compliance Intelligence, Tool Development and Real-World Application, Explainability and Trust in AI Systems, Human-in-the-Loop Approaches, Evaluation and Benchmarking. The analysis examines their presence across stages, highlights the rise of LLM-assisted rule discovery, and identifies assurance practices. The review presents a stage-based gap analysis, an evidence-based agenda for interpretable, auditable, multimodal ACC, and a reproducible method for maintaining a living review over time.

Contents

1. Introduction 2
2. Research landscape and analytical framework 3
   2.1. Research landscape 3
   2.2. Gaps in prior research (pre-2022 baseline) 4
   2.3. Stage-based analytical framework 4
3. Research methodology 5
   3.1. Search process 5
   3.2. Search strategy and query design 5
   3.3. AI-assisted discovery and provenance safeguards 5
   3.4. Screening and eligibility criteria 5
   3.5. Semantic filtering and scoring 8
   3.6. Evidence matrix construction and validation 8
4. Pre-synthesis profiling 10
   4.1. Depth versus breadth in thematic engagement 10
   4.2. Structural segmentation revealed by the heatmap 10
   4.3. Peaks, plateaus and performance bottleneck 10
   4.4. Structural fragmentation in paper–theme 10
   4.5. Ceiling effects and thematic development profiles 10
5. Stage-based synthesis 13
   5.1. Rule interpretation 13
      5.1.1. Multimodal information extraction 13
      5.1.2. Formalisation of regulatory text 13

* Corresponding author.
E-mail addresses: youssef.senousy@mail.bcu.ac.uk (Y. Senousy), Franco.Cheung@bcu.ac.uk (F. Cheung), beachth@cardiff.ac.uk (T. Beach), Ogerta.Elezaj@bcu.ac.uk (O. Elezaj), Edlira.Vakaj@bcu.ac.uk (E. Vakaj).

https://doi.org/10.1016/j.autcon.2026.107165
Received 17 November 2025; Received in revised form 15 July 2026; Accepted 21 July 2026
Available online 4 August 2026
0926-5805/© 2026 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

5.2. Model preparation ... 13
5.2.1. Semantic alignment with BIM/IFC ... 13
5.2.2. Integration of ontologies and knowledge graphs ... 14
5.3. Rule execution ... 15
5.3.1. Rule representation and reasoning ... 15
5.3.2. Model–driven compliance intelligence ... 15
5.4. Reporting ... 16
5.5. Decision support ... 16
5.5.1. Tool development and real-world application ... 16
5.6. Assurance and validation ... 17
5.6.1. Explainability and trust in AI systems ... 17
5.6.2. Human-in-the-loop approaches ... 17
5.6.3. Evaluation and benchmarking ... 17
6. Discussion and future directions ... 18
6.1. Overview and synthesis ... 18
6.2. Linking the stage-based synthesis to the research questions ... 19
6.3. Key findings ... 19
6.4. Future directions ... 19
7. Conclusion ... 20
CRediT authorship contribution statement ... 21
Declaration of competing interest ... 21
Appendix A. ChatGPT deep search protocol and provenance checks ... 21
Appendix B. AI-assisted PRISMA algorithms ... 21
B.1. Parameter calibration, and sensitivity ... 23
Data availability ... 23
References ... 32

1. Introduction

Maintaining building projects within statutory requirements is crucial for safety, sustainability, and quality in the Architecture, Engineering, and Construction (AEC) industry [1]. Before the introduction of data modelling in the industry, such as Building Information Modelling (BIM) and City Geography Markup Language (CityGML), there were few attempts to automate compliance checks [2]. Manual checking was essential during project phases, but it was prone to human error. It required substantial time and resources, often resulting in project delays and additional costs, which remain critical from a regulatory perspective [3,4]. The difficulties of the conventional manual process have increased as regulations have become more complex and interconnected with stricter sustainability standards [1].

Automated Compliance Checking (ACC) has thus emerged as a prospective alternative. In ACC, an automated process checks a digital building design to assess whether it complies with specific codes and standards, which is often supported by a rich and object-oriented data environment such as BIM [5,6]. Interoperability and trustworthy data exchange are the preconditions to successful automation. For example, Industry Foundation Classes (IFC) have been widely adopted as the open standard for data exchange in building design [7].

ACC has been developed over a significant period of time. Early adoption, such as Singapore's CORENET ePlanCheck and Australia's DesignCheck, embedded regulatory rules directly using programme coding [1,5]. This method produced fast results that are aligned to the printed regulations. However, legislative changes, even minor ones, would require the programmers to rewrite and recompile large parts of the code, which made upkeep of codes slow and costly [5]. Researchers generally agreed that this hard-coded approach could not be scaled beyond a limited set of rules for automating checks without developing better alternatives [8].

More recently, researchers have adopted ontology-based frameworks to overcome these limitations. With the development of Semantic Web technology, regulatory concepts (e.g., storey, exit width, and travel distance) are formalised as classes and properties written in Web Ontology Language (OWL) [9,10]. Constraints are typically encoded using the Semantic Web Rule Language (SWRL) or the SPARQL Protocol and RDF Query Language (SPARQL), which operates over the Resource Description Framework (RDF). This enables a reasoning engine to determine whether the relevant model elements satisfy specified requirements [11]. By decoupling knowledge from executable code, this architecture allows domain experts to revise rules directly, without editing the source files [12]. At the same time, this also enables the ontology to be reused across different projects and domains [3,13]. Converting such text into formal rules requires combining linguistic and semantic analysis with domain-specific ontologies so that the extracted rules convey the intended regulatory logic [14]. Rule capture remains difficult because regulations are often interrelated, complex, and filled with implicit references that obscure precise meaning to be identified and modelled semantically [15].

The most recent development in ACC studies applies Natural Language Processing (NLP) and Machine Learning (ML) to address the rule capturing challenge [16]. A typical workflow starts by dividing the legal text into tokens and then analysing its structure [17]. Domain-specific entity recognition is applied to identify the main objects and measurable parameters described in the regulation [18]. Building on this workflow, Zhang and El-Gohary proposed a semantic, rule-based NLP method for extracting regulatory concepts and relations from construction documents [16]. A complementary method by the same authors transformed extracted regulatory information into logic clauses suitable for automated reasoning [19]. Nuyts et al. addressed downstream validation by expressing legislative constraints as human- and machine-readable SHACL shapes for linked building data [20]. In parallel, ML models are often used to classify clause types (e.g., requirements, exceptions, and applicability) and to harmonise measurement units, which reduces the need for manual corrections.

Recent research combines NLP to extract regulatory rules, semantic web methods to represent knowledge in OWL and RDF, and data-driven models trained on regulatory examples from several jurisdictions [9,16]. This hybrid integration combines symbolic reasoning with statistical learning to support rule interpretation and validation [9]. These methods create rule bases that can be transferred across contexts and remain reliable when regulations, language, or translations change [9]. Building on that, Artificial Intelligence (AI) techniques, including large language models (LLMs) trained on regulatory corpora, can now generate candidate rules, suggest plausible preconditions, and fill gaps in BIM ontologies [21]. Experimental results from early work show

promising recall and coverage. Yet, ML-based decisions remain uncertain [22]. Hettiarachchi and et al. address the scarcity of curated regulatory data by providing an annotated corpus for rule-generation tasks [12]. Wang and El-Gohary address extraction ambiguity through deep-learning named-entity recognition and referential resolution [23]. These challenges must be addressed before LLM-generated rules can be relied upon in compliance-checking workflows [22,23].

Terminology and scope. In this paper, AI is used as an umbrella term for computational approaches supporting ACC, including symbolic (rule-based) methods, knowledge-based methods (e.g., ontologies and knowledge graphs), and machine learning methods. LLMs are treated as a subclass of machine learning, based on transformer architectures, that is increasingly used for regulatory text understanding, retrieval, and rule-generation support. This review focuses on peer-reviewed ACC studies in the AEC domain published between 2022 and 2025, with particular attention to how AI- and LLM-enabled methods are being applied across the ACC pipeline, from rule interpretation to decision support. This temporal scope was selected because earlier reviews had already established the pre-2022 foundations of rule-based, ontology-based, and BIM-centred ACC, while the period from 2022 onward captures the transition into a recent acceleration phase in which general-purpose LLMs became practically accessible and increasingly influential in regulatory interpretation, rule extraction, and AI-assisted compliance workflows. A synthesis of this period is therefore needed to capture these methodological shifts without repeating the broader historical coverage already provided by prior surveys. Accordingly, this review examines recent methodological developments within that transition period rather than attempting a direct empirical comparison between the full pre-2022 and post-2022 ACC literature.

Earlier reviews have traced the development of ACC in the AEC field. Eastman et al. (2009) described the first generation of rule-based systems within BIM [5]. Ismail et al. (2017) and Amor and Dimyadi (2021) analysed the architecture of automated compliance systems and the difficulties involved in converting regulatory rules into executable code [6,8]. Zhong et al. (2023) reviewed ontology and language based methods, while Yamusa et al. (2024) broadened the discussion to knowledge graph reasoning and early trials of LLMs [4,13]. Together, these studies established important foundations, but largely come from before the rapid growth since 2022 in AI-assisted rule extraction, multimodal regulatory understanding, and assurance focused approaches that emphasise interpretability and human oversight.

Taken together, these works provide complementary perspectives on different aspects of ACC. Building on these foundations, the review brings them together in a stage-based analytical structure comprising five primary process stages and one cross-cutting assurance layer. **Rule interpretation** extracts and formalises regulatory meaning from text, tables, figures, and cross-references. **Model preparation** prepares and semantically aligns BIM/IFC data so that relevant regulatory concepts can be queried and checked. **Rule execution** is the stage in which computable rules are operationalised and run against prepared model data through an execution mechanism to produce raw compliance verdicts and machine-level traces. **Reporting** communicates those outcomes through violations, evidence links, affected model elements, clause references, and rule traces in forms that can be inspected and verified. **Decision support** uses those reported outputs to support human review, corrective action, redesign, approval, override, or workflow action. **Assurance and validation** are treated as cross-cutting properties spanning all stages rather than as a separate stage. This stage-based view of the ACC pipeline is illustrated in Fig. 1. Within this structure, mapped technical themes are used as analytical lenses to examine how different tasks, representation choices, techniques, and assurance-related concerns appear across the pipeline. Across this process, responsibilities are distributed among key actors, including regulators or standards bodies, domain experts, BIM authors or modellers, tool developers, and compliance practitioners or auditors.

Earlier AI-oriented ACC work tended to concentrate on limited rule representation patterns and on linking BIM models to rule ontologies, with fewer reviews systematically comparing developments across the full pipeline. Since 2022, ACC research has explored LLM-assisted rule extraction and multimodal methods that process regulatory text alongside tables and figures. This expansion has also introduced methodological diversity that can hinder consolidation and cross-study comparison. Accordingly, this review synthesises recent ACC studies, identifies methodological trends across the pipeline within the 2022–2025 review window, and examines the emerging influence of AI and semantic modelling on rule extraction, semantic alignment, and explainability. These aims motivate the following research questions, which frame the study.

1. RQ1. What recent methodological developments can be observed in ACC research between 2022 and 2025 in the era of AI and LLMs, in terms of the main techniques, targeted ACC tasks, and evaluation/validation practices reported in the reviewed studies?
2. RQ2. How do the mapped technical themes manifest across the main ACC stages in terms of targeted tasks, representation choices, applied techniques, and emerging maturity patterns?
3. RQ3. What are the challenges that remain in achieving interoperability, explainability, and validation across ACC systems, and how have recent works addressed them?

This review builds on the most recent studies through five main steps. **(1)** It compiles and analyses a curated corpus of peer-reviewed ACC studies published between 2022 and 2025, providing a focused evidence base for examining recent AI- and LLM-enabled developments across the ACC pipeline. **(2)** It then applies computational coding and semantic mapping, using semantic embeddings and cosine similarity to identify recurring technical patterns and quantitative links across the reviewed studies. This supports an evidence-based and replicable view of how recent ACC research is structured. **(3)** It follows a Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) workflow to guide literature identification, screening, and study selection in a transparent and traceable manner. **(4)** It organises the synthesis around the ACC stages as the primary analytical structure, while using mapped technical themes as supporting analytical lenses to examine tasks, representation choices, techniques, and validation concerns within and across those stages. **(5)** It combines automated thematic mapping with manual validation to produce a transparent and repeatable synthesis. This approach is needed to provide a current, unbiased, and reproducible synthesis that clarifies emerging directions in AI-assisted compliance checking and establishes a foundation for future cross-disciplinary work.

2. Research landscape and analytical framework

ACC research has evolved from early rule-based approaches towards increasingly integrated semantic, data-driven, and AI-assisted methods. The analysis first outlines this development and identifies the main gaps inherited from earlier work before defining the stage-based analytical framework adopted in the review.

2.1. Research landscape

Initially, researchers began extracting rule-relevant information from narrative regulatory text. The ability to capture rule-relevant clauses and quantitative constraints from unstructured language demonstrated how compliance systems could structure regulatory text for downstream checking [16].

Building on this progress, studies addressed the gap between regulatory intent and digital models by aligning regulatory meaning with BIM or IFC representations. The accuracy of compliance checks relies

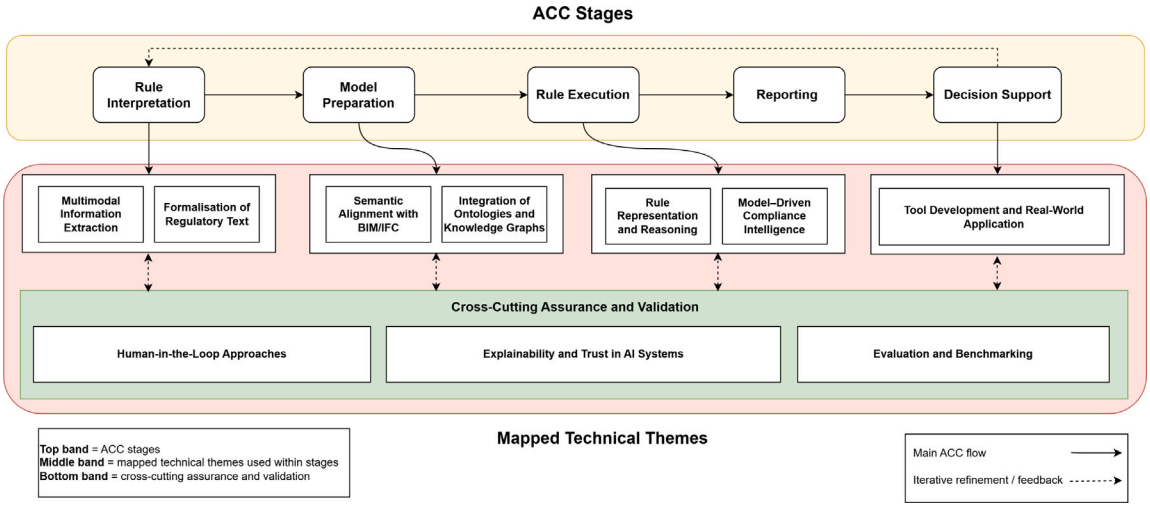


Fig. 1. Stage-based ACC process and mapped technical themes used in the review.

on a close alignment between the semantics of rules and the BIM objects. This may require domain enrichment or schema extensions for non-standard attributes and geometric constraints [9,24].

Once extraction and alignment were in place, researchers turned to the formalisation of regulatory text as a central challenge. Template-based and semantic approaches aimed to transform legal language into a form that the rule engines could process. Nonetheless, the challenges posed by legal categories and linguistic ambiguity have remained constant, particularly in provisions that could not be easily codified and in clauses that resisted clear logical mapping [25,26].

From there, focus shifted to the representation and reasoning of rules. Hard-coded procedural scripts gave way to declarative rule languages. Formal reasoning over data linked to BIM was actively explored, leading to the popularity of SPARQL, SHACL, and SWRL. Early studies extracted logic-based representations, followed by later work on OWL rule sets for IFC classes. Rule checking was structured into four functional stages: rule interpretation, building-model preparation, rule execution, and reporting of checking results [5]. Rules were also classified according to their information and computational requirements [27]. Despite their expressiveness, these approaches often required expert configuration and lacked transparency in practical use.

To address this, ontologies and knowledge graphs have become more popular for representing regulatory knowledge in a reusable and comprehensible form. Such structures allow regulatory concepts and their relationships to be defined in modular, shareable forms and integrated into project tools. RDF/OWL building data have been connected with machine-readable rules through Semantic Web approaches [9]. In parallel, visual rule-authoring languages have made rule representation more accessible to domain experts [2].

Recent studies have shown increasing interest in combining structured reasoning methods with LLMs to interpret and adapt regulatory rules for specific projects. This integration enables compliance systems to evolve beyond static rule sets and support more flexible, query-based interaction [28].

Recognising the evolving limits of full automation in these intelligent systems, Human-in-the-Loop (HITL) interfaces allow experts to review ambiguous cases and refine rules through small iterations [29]. Broader AEC research also highlights the ethical importance of human oversight in AI-supported decisions [30]. Similarly, Explainable AI (XAI) feedback systems address the growing need to understand and trust AI-based compliance tools that support auditing and accountability [31,32].

Finally, benchmarking and evaluation enhance methodological rigour by comparing validation support across technologies and extending assessment beyond binary outcomes [7,33]. Moreover, tool development and deployment boost the usability of research prototypes by making them operational and integrating workflows within the AEC industry [34,35].

2.2. Gaps in prior research (pre-2022 baseline)

Despite significant technical progress, large-scale use of ACC in daily practice remains limited. Four connected challenges dominate the current discussion:

- **Rule-capture bottleneck:** Converting lengthy and sometimes ambiguous legal text into machine-readable rules still relies on expert manual effort [22]. NLP systems can now extract entities, thresholds, and logical conditions with better accuracy, but they remain only partial aids [16,19].
- **Limited semantic interoperability:** Common BIM models mostly describe geometry and a few attributes. They seldom include the functional or contextual meaning needed for complete, cross-disciplinary checks. Links to external knowledge bases are still uncommon [7,9].
- **Domain-specific operational solutions:** Most commercial and academic prototypes focus on a single regulatory area, such as fire safety, energy, or accessibility. Their rule sets cannot easily be adapted to other fields without significant redevelopment [1,5].
- **Opaque inference engines:** The reasoning components in leading checkers, such as Solibri or Revit plug-ins, are closed-source. Their proprietary design blocks independent review of reasoning steps, reducing user trust [36,37].

These established limitations serve as a reference point for interpreting the 2022–2025 literature. They identify recurring bottlenecks that are examined again in the stage-based synthesis (Section 5), showing where recent work makes incremental progress and where key constraints remain unresolved.

2.3. Stage-based analytical framework

Building on the five-stage ACC pipeline introduced in Section 1, the reviewed studies were examined in relation to the stage or stages they addressed. Through iterative comparison across these stage-based

groupings, recurrent methodological patterns were identified and consolidated into ten mapped technical themes. These themes are shown in Fig. 1 and summarised in Table 1. They are not treated as a separate classification window. Rather, they serve as supporting analytical lenses used to examine how different tasks, representation choices, techniques, and assurance-related concerns appear within and across the ACC pipeline. The first three stages map directly to paired technical themes, while decision support is represented by the theme of tool development and real-world application.

Reporting is treated in this review as a distinct outcome-facing stage concerned with how compliance results, evidence links, clause references, and rule traces are communicated. Because this function is defined primarily by output form and traceability rather than by a separate recurring technical theme, it is analysed directly in the stage-based synthesis rather than represented here as an additional mapped theme.

3. Research methodology

To ensure a transparent and reproducible review, the methodology combines a PRISMA-based search and screening process with AI-assisted discovery, provenance safeguards, semantic filtering, and evidence-matrix validation.

3.1. Search process

A PRISMA-based search strategy was used across four complementary research sources to cover the field comprehensively [55]. Bibliographic records from Scopus, Web of Science, and IEEE Xplore captured peer-reviewed research in the AEC and computing domains. These were supplemented with an AI assisted discovery step using ChatGPT Deep Search to surface additional candidate studies and reduce the risk of missing recent or differently termed work, with inclusion subject to the same screening and verification controls described in Section 3.3. The review covers publications from 1 January 2022 to 28 July 2025.

An AI-assisted step was incorporated into the PRISMA workflow to support literature discovery and corpus organisation. All records were subsequently subject to the same screening, verification, and full-text assessment procedures.

3.2. Search strategy and query design

Given the cross-disciplinary scope and deductive focus of ACC themes, an eight-cycle iterative query scheme was applied, as summarised in Fig. 2. Each cycle used the core phrase "Automated Compliance Checking" restricted to the review window (2022–2025) and added one thematic lens (e.g., NLP, BIM, IFC, rule-based methods, ontology, knowledge graphs, LLMs, and AI) to capture domain-specific and emerging technologies. The per-cycle keywords are disclosed to ensure the search process is transparent, auditable, and readily reproducible, and to show how the retrieval scope was deliberately expanded to reflect the main technical paradigms of ACC and adjacent terms that are indexed differently across AEC and AI publication venues. In each cycle, the lens terms were combined with the ACC anchor term (for example, in Cycle 8, "Artificial Intelligence" OR "AI" is treated as one lens and then coupled with ACC). This staged formulation was adopted instead of a single broad OR query to maintain ACC specificity, limit irrelevant returns, and accommodate vocabulary drift across sub-communities. To preserve precision, the database search did not use a fully synonym-expanded query around the domain term, but instead used "Automated Compliance Checking" as a precision-oriented anchor across eight thematic query cycles, complemented by AI-assisted discovery to mitigate terminology drift. These query cycles were used to assemble the initial candidate corpus, after which thematic relevance was assessed in a separate stage through cosine-similarity ranking against the predefined theme descriptors and subsequent manual full-text review.

3.3. AI-assisted discovery and provenance safeguards

To reduce the risk of overlooking relevant studies within a rapidly evolving and cross-disciplinary ACC literature, an LLM-enabled discovery step was used solely to broaden recall, while ensuring that all inclusion decisions remained grounded in auditable bibliographic provenance and full-text assessment. In this capacity, the AI tool serves as a recall and coverage safeguard that complements conventional database searching, rather than replacing it.

ChatGPT Deep Search was used as an auxiliary discovery tool to surface additional candidate studies, rather than as the primary bibliographic source. As an externally hosted service, its retrieval behaviour and coverage are not fully transparent or controllable and may change over time. Moreover, results are often derived from publisher-facing metadata rather than consistent full-text access. Deep Search was therefore used to improve recall by mitigating indexing lag and terminology drift, and any candidates were incorporated into the corpus only after passing the same de-duplication, eligibility screening, provenance verification, and full text assessment applied to database records. The Scopus, Web of Science, and IEEE Xplore exports remained the bibliographic backbone of the final dataset.

Because Cycle 1 was executed with an intentionally inclusive scope, Deep Search may surface outputs that fall outside conventional indexing channels, including grey literature such as SSRN or arXiv preprints. In Cycles 2–8, prompts restricted retrieval to peer-reviewed journal and conference venues; however, Deep Search may still return non-publisher access points hosted on repositories or aggregator sites even for otherwise peer-reviewed works. To manage provenance risks, including unverifiable citations and material that has not undergone peer review, each Deep Search output was treated strictly as a candidate lead. A record was retained only when its bibliographic identity could be corroborated through stable identifiers (DOI where available) and/or trusted publisher or indexing entries, when the outlet satisfied the study's peer-reviewed eligibility criteria, and when the full text was accessible for evaluation. Records lacking verifiable identifiers or reliable source traces, or remaining preprint only without a confirmable peer-reviewed counterpart, were excluded during screening. This procedure improves recall while ensuring that inclusion decisions remain anchored in auditable provenance and full text appraisal. The ChatGPT Deep Search cycles were executed on 28 July 2025; the fixed prompt protocol, query log, and archived CSV exports are reported in Appendix A.

3.4. Screening and eligibility criteria

The bibliographic exports from the four data sources differed in both file format and metadata structure. All retrieved data were converted to UTF-8-encoded CSV files to standardise export formats (e.g., Excel from Web of Science and CSV from Scopus and IEEE Xplore). Field normalisation, correction of encoding artefacts, and removal of intra-source duplicates were performed using a custom Python script to create a clean unified dataset for cross-source integration.

Next, the four source-specific files were merged into a single corpus. Database-specific column names, such as *Document Title* in IEEE Xplore and *Article Title* in Web of Science, were mapped to a standard five-column schema: *Author, Title, Year, Keywords, Abstract*. After normalisation, all records were concatenated and cross-source duplicates were removed automatically, resulting in 272 unique records. A subsequent manual clean-up step resolved residual duplicates and incomplete or inconsistent metadata, without applying relevance-based exclusions, yielding 205 records for screening.

Table 1
Mapped technical themes used as analytical lenses within the stage-based ACC framework.

Primary stage locus	Theme	Description	Keywords	Representative work
Themes primarily addressed during regulation digitisation and preparation				
Rule interpretation	Multimodal information extraction	Extracting regulatory data from text, images, tables, and diagrams using NLP and computer vision, then structuring it for downstream reasoning and compliance checking.	Natural Language Processing, NLP, multimodal extraction, document AI, OCR, table extraction, diagram understanding, layout analysis, visual document parsing, doc-understanding models, PDF to structured, document layout modelling, semantic parsing, rule mining from tables, multimodal NLP, vision language models, image text alignment, semantic segmentation, CV fusion.	Zhang & El-Gohary [19], [16], Zhong et al. [38], Wu et al. [39], Fang et al. [40], Xue & Zhang [41]
	Formalisation of regulatory text	Building regulations are transformed from prose and tables to precise, logic-ready structures by decomposing requirements, capturing parameters and thresholds, modelling applicability conditions and exceptions, resolving cross-references and dependencies, and encoding them in a traceable logic representation with tree and graph views for automated compliance reasoning.	text formalisation, clause structuring, logic based representation, predicate/production rules, syntactic/semantic NLP, parameter capture, threshold extraction, applicability conditions, exception modelling, dependency modelling, cross-reference resolution, rule atomisation, source traceability, computable rules, reasoning-ready constraints.	Zhang & El-Gohary [16], [19], [42]
Themes for BIM readiness and semantic grounding				
Model preparation	Semantic alignment with BIM/IFC	Establishing clear and computable links between regulatory requirements and IFC based BIM data by aligning rules with specific entities, properties, and relationships, enabling accurate querying and automated compliance checking through schema mapping and BIM aware rule execution.	Semantic alignment, IFC mapping, rule to model alignment, entity and property mapping, property set alignment, BIMQL queries, constraint to IFC property mapping, rule to BIM binding, BPMN and DMN with BIM, SHACL, SPARQL, compliance oriented object model, IFC schema extension, spatial structure and topology, traceability to model elements, model checking.	Zhang & El-Gohary [16], [19], [24], Mazairac & Beetz [36]
	Integration of ontologies and knowledge graphs	Ontologies and knowledge graphs used to formalise regulatory concepts, domain entities, and their interactions, allowing for semantic reasoning, rule-based compliance verification, and data integration across multiple sources. These structures enable automated inference, consistency checking using formal languages like as OWL, SWRL, and SPARQL.	ontology, ontology modelling, domain ontologies, knowledge graph, semantic reasoning, rule based compliance checking, ontology-driven integration, OWL, SWRL, RDF, semantic interoperability, domain semantics, ontology–BIM integration, visual-semantic reasoning, logic-based rule encoding, semantic annotation, multi source data fusion.	Zhang & El-Gohary [43], Zhong et al. [44], Wu et al. [39], Fang et al. [40]
Themes for executable compliance reasoning				
Rule execution	Rule representation and reasoning	Creating and implementing formal representations of regulatory requirements in logic-based, semantic, or visual languages to facilitate automated compliance reasoning. These approaches include first-order logic clauses, rule trees, visual rule languages, decision models, and semantic rule formats, all of which enable uniform, visible, and executable rule assessment within BIM and regulatory systems.	Rule representation, logic based modelling, first order logic, logic clauses, tree-based rule structures, visual rule language, DMN, BPMN, semantic rule encoding, RuleML, rule classification, rule schema, logic programming, compliance reasoning, rule evaluation, reasoning engine.	Solihin & Eastman [27], Zhang & El-Gohary [19], [43], Boley et al. [45], Zhang [42], Preidel & Borrmann [46], Häußler et al. [47]
	Model-driven compliance intelligence	Using large language models, deep learning, and retrieval-augmented generation to automate compliance understanding, rule extraction, and regulatory interpretation. These approaches combine semantic retrieval, ontology grounding, and few-shot learning to improve adaptability, eliminate manual rule authoring, and enable domain-specific compliance reasoning in BIM and workflows.	Large language models, LLM, GPT, BERT, ChatGPT, retrieval augmented generation, RAG, RAG-graph, neural-symbolic, prompt engineering, few-shot learning, one-shot learning, compliance automation, text-to-rule generation, deep learning, semantic retrieval, vector embedding, compliance AI pipeline, fine-tuned transformer, domain adaptation, hybrid AI frameworks.	Yamusa et al. [4], Chen et al. [28], Wan et al. [21], Kampelopoulos et al. [48]
Theme for decision support, deployment, and practical application				
Decision support	Tool development and Real-World application	Designing, implementing, and validating practical systems for ACC within real-world environments.	Compliance tool, BIM integration, IFC-based checking, domain-specific language, tool validation, automated BIM checking, knowledge-driven tool, generative design, workflow automation, pilot project, real-world implementation, case study evaluation, lifecycle, UI, adoption.	Mazairac & Beetz [36], Preidel & Borrmann [46], Fang et al. [40], Sydora & Stroulia [49], Häußler et al. [47]

(*continued on next page*)

Table 1 (*continued*).

Themes for trust, oversight, and evaluation				
Cross-cutting across all stages	Explainability and trust in AI systems	Building compliance checking systems in a way that offer interpretable, user understandable outputs to help develop trust with the practitioners, auditors, and regulators. This involves application of visual logic models, transparent rule structures, and intuitive representations that allow users to trace rule decisions back to source regulations.	Explainable compliance, tree based visualisation, user understandable logic, interpretable rules, regulatory traceability, transparent reasoning, compliance explainability, trust in AI, AI-assisted decision-making, visual logic modelling, rule transparency, provenance, auditability.	Zou et al. [1], Amor & Dimyadi [8], Zhang [42], Beach et al. [10]
	Human-in-the-loop approaches	Embedding human expertise into automated compliance workflows for supervision, correction, and refinement. These methods keep experts involved through review, annotation, or intervention stages to guide machine reasoning, improve accuracy, and ensure accountable decision-making.	HITL, Human-in-the-Loop, hybrid AI-human, interactive compliance checking, user-in-the-loop, expert-in-the-loop, manual review, expert validation, human oversight, participatory rule authoring, semi-automated verification, human feedback loop, collaborative AI, active learning, annotation loop, manual override, quality assurance, expert correction.	Beach et al. [10], Zou et al. [1], [50]
	Evaluation and benchmarking	Evaluating the performance, accuracy, usability, and readiness of automated compliance checking systems through comparative studies, reviews, and standardised metrics.	Benchmarking, evaluation metrics, performance assessment, comparative analysis, accuracy, precision, recall, F1-score, rule coverage, tool maturity, usability testing, adoption readiness, validation frameworks, evaluation criteria, compliance testing metrics, dataset, generalisation metrics.	Dimyadi & Amor [51], Ismail et al. [6], Pauwels et al. [52], Beach et al. [10], Amor & Dimyadi [8], Fuchs [53], Li et al. [54]

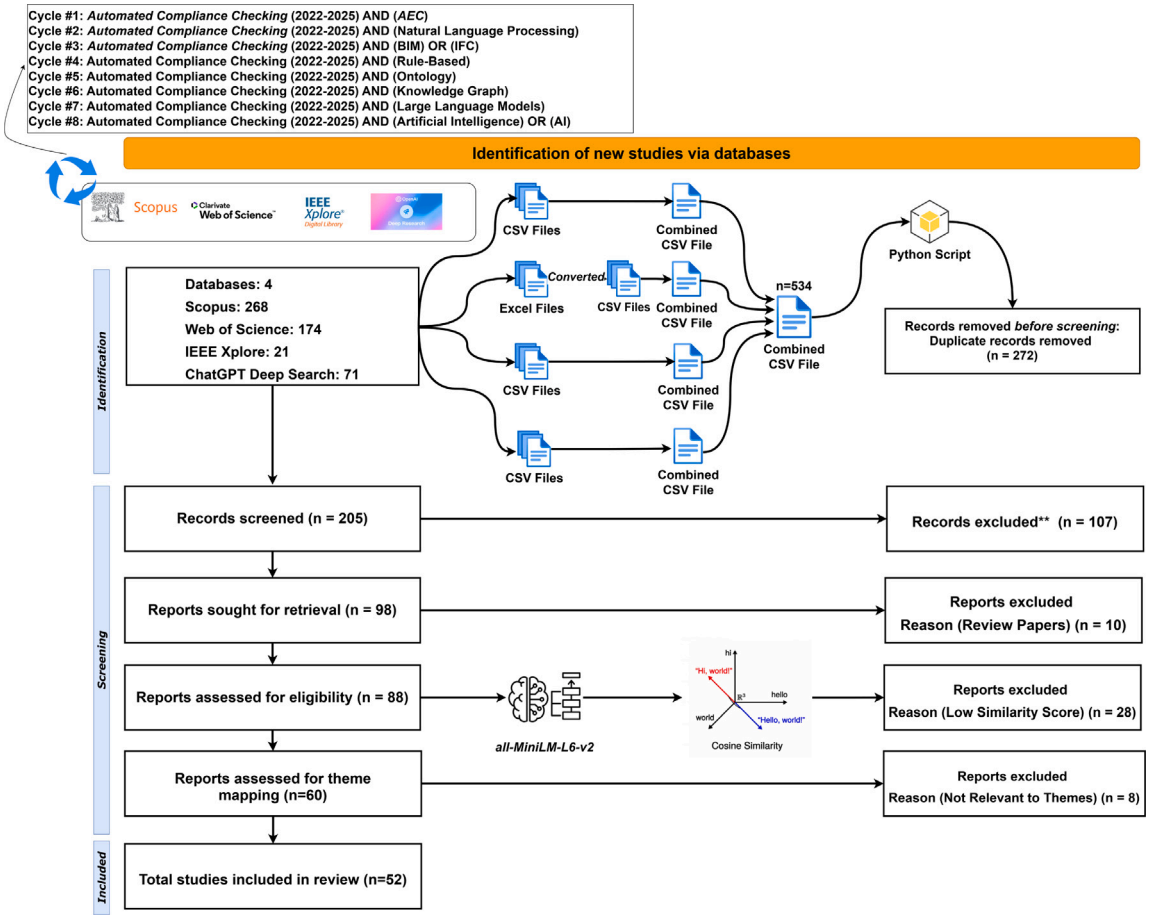


Fig. 2. AI-assisted PRISMA workflow.

Eligibility criteria (inclusion and exclusion). Eligibility criteria were defined *a priori* to ensure consistent study selection. Studies were included if they: (i) addressed ACC in the AEC domain (including BIM/IFC-enabled workflows where applicable), (ii) were peer-reviewed journal or conference publications, (iii) were published within the review window (2022–2025), (iv) proposed, evaluated, or substantively analysed an ACC method, system, or pipeline, and (v) had accessible full text in English. Studies were excluded if they were editorials or other non-peer-reviewed outputs.

Title/abstract screening. Titles and abstracts were screened against the predefined eligibility criteria to identify records for full-text assessment.

For each record, the screening decision was logged as *Selected* or *Not selected* to maintain consistency and provide an auditable trail. This stage reduced the screened set to 98 records for full-text assessment.

Quality safeguards and risk-of-bias considerations. A formal risk-of-bias tool was not applied at the individual-study level because ACC studies are highly heterogeneous in terms of objectives, datasets, and evaluation designs, and standard instruments are not readily transferable to this type of technical evidence synthesis. Instead, methodological credibility was supported by predefined inclusion and exclusion criteria and a documented screening process. These criteria served as formal screening guidelines and were applied consistently across title and abstract screening, full-text review, and theme assignment using a single structured Excel screening sheet (`AllSources_Combined_Clean_Final.xlsx`) stored in the project `GitHub` repository https://github.com/ysenousy/ACC-N-ACE. Screening and theme assignment were conducted by the first author, and all decisions, including inclusion decisions and full-text exclusions, were systematically documented to ensure consistency, traceability, and an auditable review process. Accordingly, the findings should be interpreted as a stage-based synthesis informed by thematic mapping of reported evidence rather than a quality-weighted meta-analysis.

Screening summary. Overall, 534 records were retrieved (Scopus: 268; Web of Science: 174; IEEE Xplore: 21; ChatGPT Deep Search: 71) and reduced to 272 unique records after de-duplication; 205 records were screened by title and abstract, yielding 98 full-text articles, of which 10 review papers were excluded, resulting in 88 primary studies for synthesis.

3.5. Semantic filtering and scoring

After all PDFs in the article dataset were finalised and downloaded, each was converted into a machine-readable text file containing the main body, tables, and Optical Character Recognition (OCR) extracted text from figures and diagrams. The resulting text representations were embedded using the *all-MiniLM-L6-v2* model from Sentence-Transformers, based on the architecture of Reimers and Gurevych [56]. Models such as *SPECTER2* and *SciNCL* are specifically designed for scientific document-level embeddings and can provide stronger paper-level relevance modelling in related evaluation settings; however, some *SPECTER2* variants and citation-informed training pipelines introduce additional parameters and runtime costs, which can become material when embedding large corpora [57,58]. Accordingly, *MiniLM* was selected as an efficient initial screening model to support scalable similarity ranking, with final inclusion decisions based on manual full-text review [59]. This process produced 384-dimensional sentence embeddings suitable for semantic comparison. Cosine similarity was then calculated to determine how closely each vector corresponded to a curated set of theme-specific reference embeddings constructed from the predefined theme descriptors and associated keyword vocabularies reported in Table 1, following a standard method for comparing data in high-dimensional space [60]. The articles were subsequently ranked by similarity score to indicate their degree of alignment with each thematic category. This step generated an automatically prioritised reading list and helped identify conceptually related work that keyword filters might otherwise overlook. Fig. 3 illustrates the semantic filtering and scoring workflow within the research methodology.

The workflow for this stage is formalised in two algorithms. `Algorithm 1` iterates through all PDFs in the collection, extracts content (body text, tables, and OCR-derived figure text), and compiles it into a structured text representation for each article. `Algorithm 2` then embeds candidate sentences using *all-MiniLM-L6-v2* and computes cosine similarity against reference vectors representing the review themes. Similarity is aggregated across document sections using a section-weight vector to support triage and thematic profiling. The resulting scores produce a repeatable ranking of papers by thematic relevance for subsequent manual synthesis. Appendix B provides the full pseudo-code for both algorithms together with the parameter rationale and sensitivity analysis (tested grid, overlap/Jaccard computation, and paper- and theme-level outcomes).

Collecting these outputs from all documents produces a thematic similarity matrix that maps the corpus across ten themes and identifies the evidence sentences supporting each theme. This structure supports both quantitative synthesis and qualitative review.

Although semantic filtering and ranking provide a scalable and thorough approach for mapping articles to thematic categories, the method has certain limitations. *MiniLM* sentence embeddings are computationally efficient but can miss nuanced sentence structure in long or complex sentences, as the model compresses linguistic information into 384-dimensional fixed-length vectors [61].

Furthermore, cosine similarity measures only the angular proximity between embedding vectors and does not account for syntactic structure, logical flow, or wider discourse context [62]. This results in a coarse assessment of thematic consistency rather than a critical interpretation. Sentences are evaluated independently, without considering immediate context, paragraph structure, or inter-sentence connections that may carry major conceptual links [62].

The section-weighting applied to body text, tables, and OCR-derived figure text and the threshold-based labelling (*direct* vs. *partial*) are design choices used to support *triage and thematic profiling*, rather than to replace full-text eligibility assessment. To mitigate the risk of parameter sensitivity, the parameter rationale is documented and a sensitivity analysis is reported in which thresholds and multimodal weights are varied over plausible ranges and Algorithm 2 is re-run (Appendix B.1). The resulting paper-level pool membership and set-stability measures (overlap/Jaccard), together with theme-level label totals and top-theme stability (Tables B.9 and B.10), indicate that the dominant thematic conclusions are stable across tested settings. Final inclusion decisions remain anchored in manual full-text screening with PRISMA-compatible exclusion reasons recorded for traceability.

3.6. Evidence matrix construction and validation

A summary matrix was created from 88 peer-reviewed studies to support the Stage-Based Synthesis, using the highest-ranked papers identified through similarity scoring. Representative studies shown in the summary tables were drawn from the highest-ranked papers within each theme and were retained only after direct review of the paper objectives, methods, and substantive thematic contribution. The matrix records authors, methods, tools, application areas, main contributions, thematic scores, and relevance labels. Table 2 summarises six representative studies under the theme *Formalisation of Regulatory Text*.

This theme covers research that transforms narrative building regulations into structured, machine-readable logic suitable for ACC. The table lists the authors, main methodological approach, application domain, key contribution, and the verified thematic relevance rating (R = relevant, PR = partially relevant, NR = not relevant). Relevance was confirmed through direct review of each paper's objectives and methods. For example, papers that explicitly formalise regulatory text into computable rules were classified as relevant (R), those extending the concept only partly as partially relevant (PR), and studies that are not addressed were classified as not relevant (NR). The complete matrices for all ten themes appear in Appendix B.

Table 2 is included to illustrate how the relevance rating scheme was applied in practice within the theme of Formalisation of Regulatory Text. It shows that recent studies vary in scope and in how closely they align with this theme, with contributions ranging across ontology construction, language parsing, and neural translation of rules.

Automated similarity screening reduced the corpus to 60 papers selected for detailed reading and thematic analysis, while 28 papers were excluded due to low similarity scores. As shown in Table 3, 46 papers were classified as Relevant (R) and six as Partially Relevant (PR),

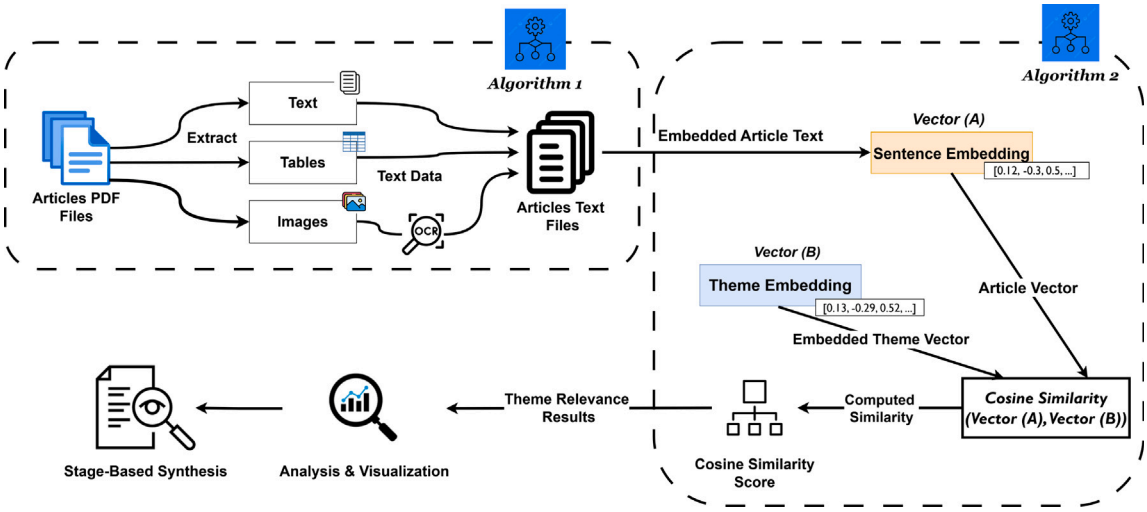


Fig. 3. Semantic filtering and scoring workflow.

Table 2
Representative studies under the theme *formalisation of regulatory text*.

Authors (Year)	Method	Domain	Main contribution	Relevance (R/PR/NR)
R. Zhang and N. El-Gohary (2022) [63]	Deep-learning encoder–decoder with attention and copying for generating machine-readable rules.	Building regulation reasoning and intelligent code generation in the AEC domain.	Introduces multi-form semantic requirement hierarchy; achieves 94% BLEU accuracy and improves semantic linking precision to 92%.	R
S. Zentgraf et al. (2022) [64]	Ontology design using Linked Open Terms (LOT) and SHACL validation for building code data.	Ontology-based representation of building code information for BIM and Linked Building Data workflows.	Develops the Interconnected Data Dictionary Ontology (IDDO) aligned with ISO 23386, enabling FAIR-compliant property management.	NR
Z. Zheng et al. (2022) [65]	Knowledge-informed NLP with ontology modelling, unsupervised alignment, and SPARQL rule generation.	Automated rule interpretation and compliance checking in BIM-based fire safety design.	Introduces FPBO ontology and unsupervised alignment with 90% accuracy; automates SPARQL rule generation for complex reasoning.	R
D. Lehoczky et al. (2022) [66]	Applies the 3S cognitive psychology model to analyse human trust in automated rule checking.	Socio-technical evaluation of user confidence in BIM-based automated rule checking.	Identifies information features affecting trust and provides design guidelines for more transparent ARC interfaces.	NR
J. Lee et al. (2023) [67]	Translates natural-language clauses into atomic sentences and pseudo-code methods for logic encoding.	Automated building permit checking under the Korean Building Act using NLP and logic-based methods.	Builds the KBIMCode and KBIMLogic tools for rule translation and demonstrates full BIM-linked compliance checking.	PR
J. Wu (2023) [68]	Combines invariant signatures, logic reasoning, and NLP in the I-SNACC framework.	BIM-based code compliance checking for spatial and quantitative requirements under IBC 2015.	Creates a modular ACC system with integrated rule generation, validation, and reasoning; achieves 95% precision and 100% recall.	R

Table 3
Summary of paper relevance classification.

Metric	Count
Total papers	60
Relevant (R)	46
Partially relevant (PR)	6
Not relevant (NR)	8

giving a total of 52 papers included in the final synthesis. The eight remaining papers were coded as Not Relevant (NR).

$$\text{Raw accuracy} = \frac{46}{60} = 0.77 \times 100 = 77\%,$$

$$\text{Weighted accuracy} = \frac{46 + 0.5 \times 6}{60} = 0.82 \times 100 = 82\%.$$

Overall, the results show that the theme mapping reached 77% Raw accuracy and 82% agreement with academic relevance evidence, forming a solid empirical basis for the stage-based synthesis.

The research methodology was explicitly designed to address the three research questions introduced in Section 1. RQ1 is addressed through the AI-assisted PRISMA search and screening process, which compiles ACC studies published between 2022 and 2025 to identify recent methodological developments reported during the emergence of AI- and LLM-enabled ACC, in terms of techniques, targeted ACC tasks, and evaluation/validation practices. RQ2 is examined through semantic embedding and cosine-similarity thematic mapping, which quantify conceptual proximity among studies and show how the mapped technical themes manifest across ACC stages in terms of targeted tasks, representation choices, applied techniques, and emerging maturity patterns. RQ3 is explored through manual validation and qualitative synthesis, which identify persistent challenges to interoperability, explainability, and validation across ACC systems, and how recent works attempt to address or mitigate them. Together, these stages establish a transparent and reproducible workflow that not only organises the literature but also provides data-driven evidence to answer each research question systematically.

Table 4
Sample from the thematic similarity matrix.

Authors	Multimodal information extraction	Semantic alignment with BIM/IFC	Formalisation of regulatory text	Rule representation and reasoning	Integration of ontologies and knowledge graphs
A. Donkers and E. Petrova 2024	0.623	0.618	0.696	0.638	0.693
A. Ismail et al. 2023	0.639	0.756	0.689	0.702	0.632
A. Karmakar and V. Delhi 2023	0.579	0.684	0.724	0.72	0.618
C. Lee et al. 2024	0.52	0.668	0.668	0.652	0.55
D. Al-Turki et al. 2024	0.626	0.702	0.733	0.666	0.576

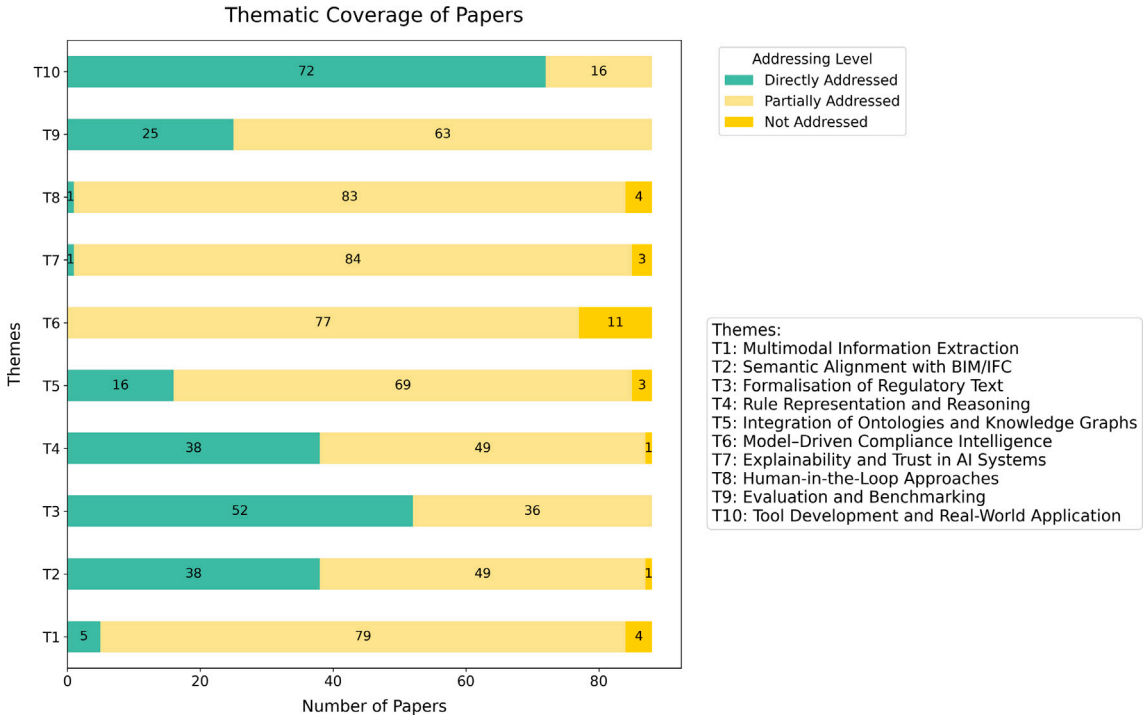


Fig. 4. Stacked horizontal bar chart.

4. Pre-synthesis profiling

Before conducting the full stage-based synthesis, it is useful to explain how the ranking matrix guided the analysis. Table 4 shows a sample of the matrix, displaying similarity scores for five example themes. The complete matrix and supporting project files are available in the same `GitHub` repository. Detailed thematic labelling and stage-based discussion of the ten cross-cutting themes are provided in the following sections.

4.1. Depth versus breadth in thematic engagement

Fig. 4 presents the overall distribution of thematic engagement across the corpus by distinguishing direct, partial, and absent coverage. It shows that direct attention is concentrated in Tool Development and Real-World Application and Formalisation of Regulatory Text, whereas most other themes are engaged more often in a supporting than a primary role.

4.2. Structural segmentation revealed by the heatmap

Fig. 5 shows how thematic intensity varies across individual papers rather than across aggregate counts alone. The distribution is visibly uneven, indicating that some themes are embedded broadly and consistently across studies, while others appear in more selective and less uniform ways.

4.3. Peaks, plateaus and performance bottleneck

Fig. 6 summarises the average semantic strength of each theme across the reviewed literature. The higher averages for Tool Development and Real-World Application and Formalisation of Regulatory Text suggest greater conceptual consolidation, while the lower averages for Model-Driven Compliance Intelligence and Human-in-the-Loop Approaches point to themes that remain less settled.

4.4. Structural fragmentation in paper–theme

Fig. 7 isolates only direct thematic matches and therefore provides a clearer view of where strong concentration actually exists. The pattern indicates that direct contributions cluster around semantic alignment, formalisation, reasoning, and practical tool development, while direct treatment of themes such as explainability and human oversight remains comparatively sparse.

4.5. Ceiling effects and thematic development profiles

Fig. 8 compares the strongest individual studies within each theme and therefore reflects the upper boundary of current thematic development. It suggests that some themes already have well-defined leading examples, whereas others still show a lower and less sharply differentiated top tier.

Taken together, these profiling views indicate that the reviewed corpus is characterised by uneven thematic maturity. They were used to support paper prioritisation and broad interpretation before the more detailed stage-based synthesis presented in Section 5.

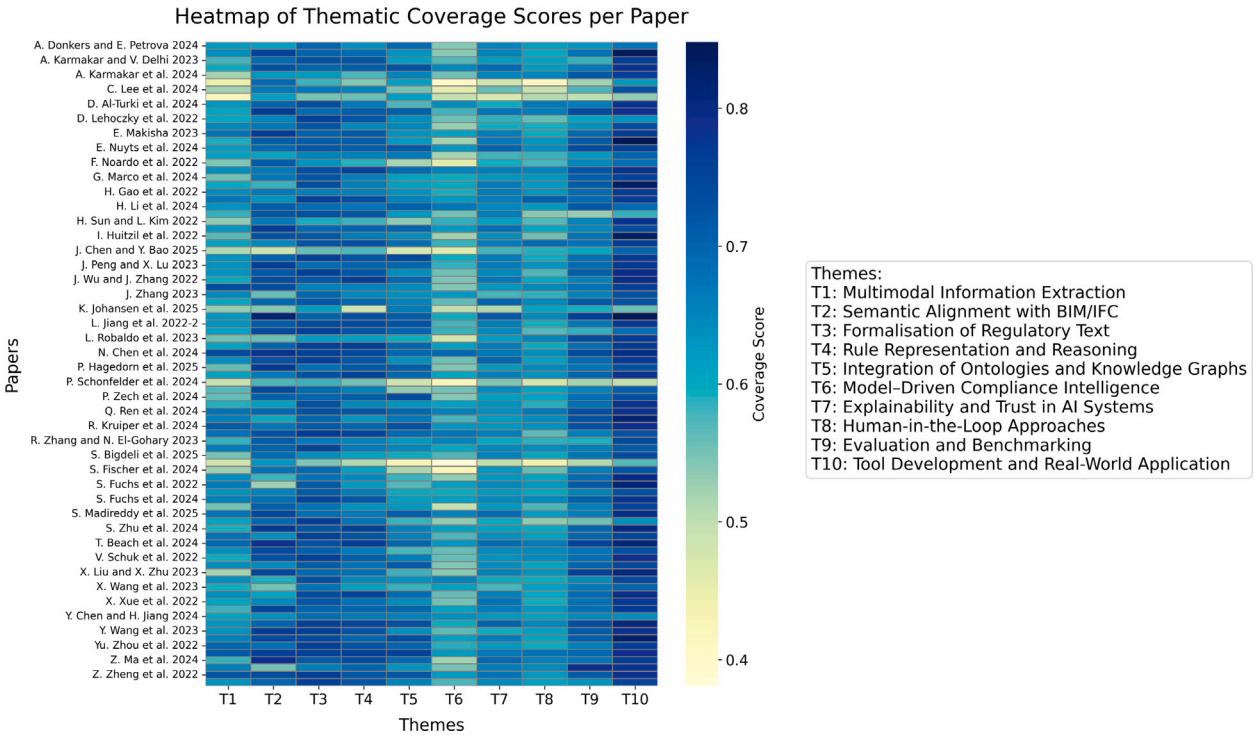


Fig. 5. Paper-level thematic similarity matrix.

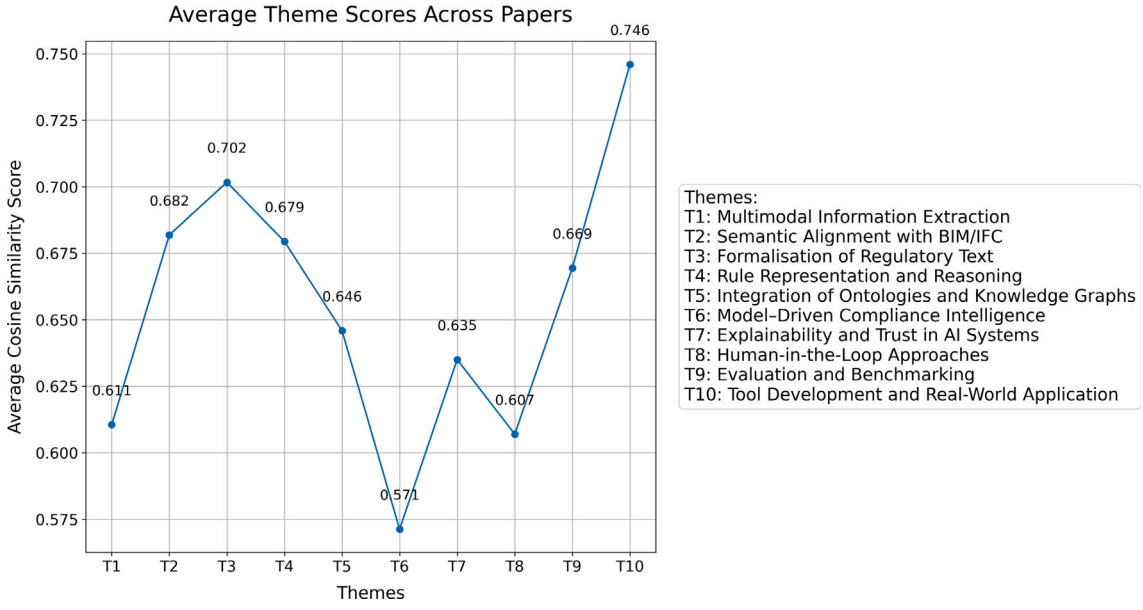


Fig. 6. Average thematic scores.

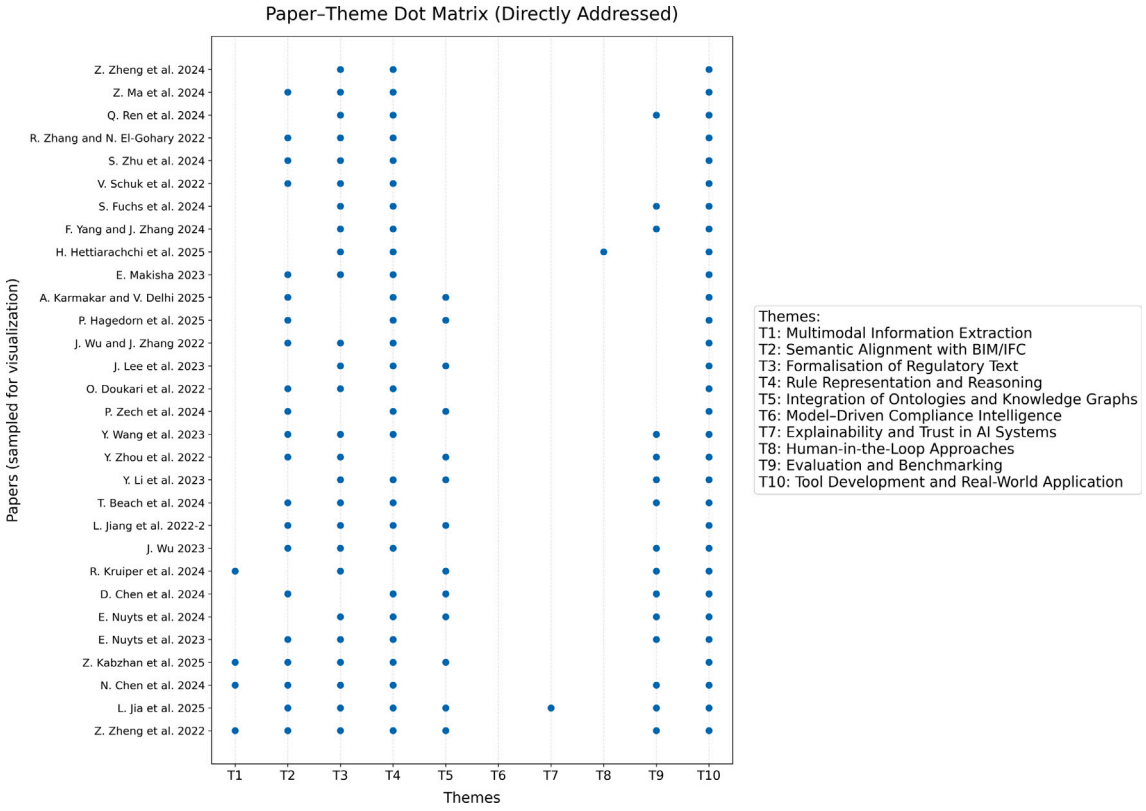


Fig. 7. Paper–themes dot-matrix.

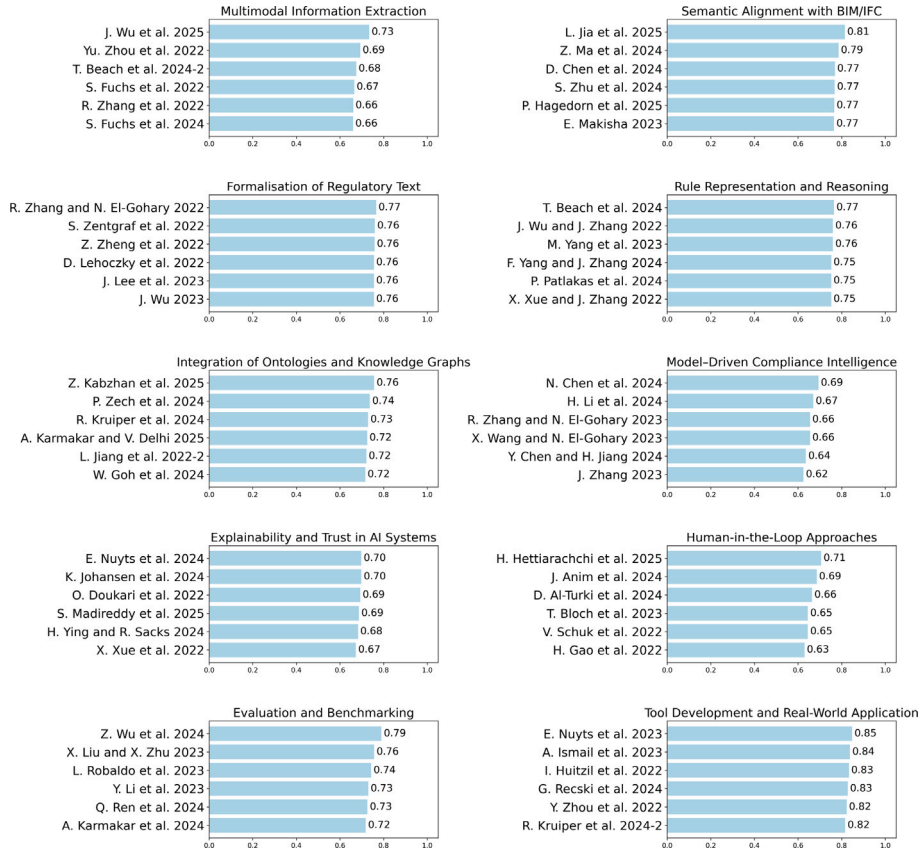


Fig. 8. Top-six ranked papers per theme.

5. Stage-based synthesis

This synthesis is organised around the five ACC stages shown in Fig. 1: (i) rule interpretation, (ii) model preparation, (iii) rule execution, (iv) reporting, and (v) decision support. These stages provide the primary analytical structure of the review. Mapped technical themes are used as supporting analytical lenses to examine the main tasks addressed in each stage, the representation choices adopted, the techniques applied, and the cross-cutting meta-properties that shape interpretability, interoperability, and validation. Accordingly, the following sections analyse each stage in a structured way, while drawing on the themes only to characterise recurring methodological patterns rather than as a separate review window.

5.1. Rule interpretation

Rule interpretation concerns the upstream ACC tasks of extracting regulatory meaning and transforming it into machine-useable structures. Within this stage, Multimodal Information Extraction and Formalisation of Regulatory Text operate as two closely connected analytical themes. The main tasks involve identifying entities, conditions, thresholds, dependencies, and supporting evidence from text and other document modalities, and then converting that material into logic-ready and traceable representations. Representation choices at this stage include syntax trees, clause hierarchies, structured rule frames, and semantic annotations, while the dominant techniques span grammar-based parsing, neural semantic parsing, information extraction, and hybrid symbolic–neural formalisation. Across these studies, the main meta-properties are traceability to source clauses, handling of ambiguity, and readiness for downstream execution and audit.

5.1.1. Multimodal information extraction

Multimodal information extraction focuses on the ACC task of converting regulatory text, tables, figures, and other evidence sources into structured content that can support downstream rule formation and checking. It aims to turn text, figures, and digital evidence into structured data that machines can process. The most advanced work still employs text-first extraction, where deep learning and grammar parsing transform clauses into logical forms. Studies using context-free grammar demonstrate high accuracy for both compound and straightforward requirements, producing labelled trees converted into rules [69]. Other research builds hierarchical models and applies neural relation extraction to split complex sentences into linked requirement units that rule engines can read [70]. Another line tests neural semantic parsing and transformer translation into legal-logic formats. Results show that adding intermediate layers can improve accuracy and speed while keeping outputs open to expert review [71,72].

Work is now extending beyond text. Recent studies explore life-cycle compliance that combines image recognition, thermography, LiDAR, IoT, and photometry to collect verifiable evidence during operation and inspection [73].

Bringing these research lines together is still unfinished. Current text pipelines rarely extract information from tables, diagrams, or layout features such as cross-referenced thresholds or schematic exceptions. Operational sensing research seldom ties its findings to text-based requirements or shared ontologies. Few datasets pair PDF pages with sentence, table, and figure annotations linked to verified rules, BIM models, and sensor data. This shortage limits the ability to make a fair comparison across methods. The field now calls for joint multimodal parsers that handle text, layout, tables, and figures together, supported by vision–language–BIM links and open benchmarks that measure extraction accuracy, cross-modal grounding, and full compliance performance. Taken together, current work shows progress in text-centred extraction, but still lacks robust multimodal representations and benchmarked cross-modal grounding needed for traceable downstream compliance use.

5.1.2. Formalisation of regulatory text

Regulatory text formalisation aims to convert descriptive building rules into precise, machine-readable logic for automated compliance reasoning. Early frameworks demonstrated how to break down clauses into atomic logical units that capture parameters, thresholds, and conditions, while maintaining traceable links to the original text [19]

Later studies expanded these ideas through rule-based parsing and the use of structured representations. Regulatory text was converted into predicate and production rules that support reasoning across clause hierarchies and dependencies [65]. A BIM-based rule-set model applied formal logic to connect structured rules with building elements and PropertySets for automated compliance checks [67]. The combination of logic programming and NLP improved interpretability by using syntactic and semantic parsing to extract constraints from complex legal texts [68].

Recent development has linked deep learning models with language generation to create intelligent building codes that automatically structure and interpret text as semantic rule hierarchies. These models apply contextual embeddings and neural generation to clarify meaning, reduce ambiguity, and produce scalable, logic-ready representations [63].

Despite these advances, key problems persist. Methods often lack semantic consistency and face interoperability gaps because formalisations rely on isolated schemas or rule expressions. Many systems still depend on manual review and fail to capture multimodal content. The absence of benchmark datasets limits reproducibility, and neural models remain opaque, raising questions about transparency. Progress will depend on ontology-linked corpora, hybrid symbolic–neural pipelines, and explainable rule-formalisation frameworks that provide traceable, adaptable, and auditable structures for real-world compliance use. Overall, the strongest progress lies in converting clause-level requirements into logic-ready structures, while interoperability, multimodal completeness, and transparent neural formalisation remain less mature.

5.2. Model preparation

Model preparation concerns the ACC tasks required to make digital building data checkable against regulatory requirements. Within this stage, Semantic Alignment with BIM/IFC and the Integration of Ontologies and Knowledge Graphs provide the main analytical lenses. The core tasks involve linking clauses to IFC entities, properties, and spatial relations, enriching BIM data with shared semantics, and preparing interoperable structures for downstream checking. Representation choices include IFC and IFCOWL models, RDF graphs, OWL ontologies, PropertySet mappings, and linked-data structures, while the main techniques involve schema mapping, ontology alignment, SPARQL/SHACL/SWRL-based integration, and semantic enrichment workflows. The dominant meta-properties at this stage are interoperability, portability across tools and IFC versions, traceability to model elements, and robustness of rule-to-model grounding.

5.2.1. Semantic alignment with BIM/IFC

Semantic alignment links regulatory requirements to IFC-based BIM data by mapping clauses to IFC entities, properties, and spatial relations. This connection allows reasoning tools to run queries directly on BIM models. One method involves building a domain code ontology, converting IFC data into IFCOWL, and connecting the two through RDF mappings and SPARQL rule templates on Apache Jena (Apache Jena is an open source Semantic Web framework for Java). This workflow establishes complete alignment from textual clauses to IFC-based checks and proves its effectiveness through case studies [74].

Related studies formalise quality standards in OWL and express rules in SWRL. These rules are then executed directly on IFC models through inference, demonstrating that ontology-based systems can automatically verify models within the IFC environment [75].

At the process level, OntoBPR (ontology-based building permit review) combines several domain ontologies with ICDD (Information

Table 5
Representative semantic-alignment approaches and their interoperability exposure across IFC-centred ACC workflows.

Study	Alignment approach	IFC anchoring	Interoperability exposure	Main limitation
Jia et al. (2025) [74]	Domain ontology, IFCOWL, RDF/SPARQL templates	Explicit IFC-to-IFCOWL mapping	Strong semantic traceability and clause-to-model linkage; more portable than tool-specific implementations	Still depends on export fidelity and mapping completeness across authoring environments
Ma et al. (2024) [75]	OWL ontology, SWRL rule execution on IFC models	IFC entities mapped into ontology	Effective for ontology-driven checking within a formal semantic environment	Sensitive to schema/property availability and ontology-mapping effort
Hagedorn et al. (2025) [76]	OntoBPR with ICDD, SHACL, IFCOWL, BOT, and OPM	Linked-data and permit-workflow integration	Strong standards-based validation and cross-document traceability	Requires coordinated modelling, document, and workflow conventions to remain interoperable
Makisha (2023) [78]	RuleML-based rule generation with IFC terminology mapping	Explicit IFC 4.2 terminology mapping	Useful for structured text-to-IFC linkage and version-explicit formalisation	More manual and potentially version-specific in practice
Zhu et al. (2025) [77]	Knowledge graph, Revit plug-in, and SQL rule base	Native BIM/tool-level checking rather than full IFCOWL, SHACL, and SPARQL translation	Practical within the authoring environment, but less portable across tools and exports	More exposed to authoring-tool conventions and PropertySet variability

Container for Linked Document Delivery) containers and SHACL to support digital permit workflows. This framework shows how IFC and IFCOWL data, along with building ontologies such as BOT (Building Topology Ontology) and OPM (Ontology for Property Management), can be linked to rule constraints to enable transparent, standards-based validation in permitting [76].

Other studies partly advance IFC-level alignment. A knowledge graph-based BIM quality checker structures modelling and delivery standards, and carries out automated checks through a Revit plug-in [77]. It has not yet transferred constraints to IFCOWL classes, SHACL, or SPARQL queries; however, it improves rule traceability and flexibility [77]. Simultaneously, another research on regulatory text translation explores which clauses can be automated and suggests writing patterns to avoid uncheckable rules [78].

However, real-world semantic alignment remains more fragile than these architectures imply. In practice, interoperability is constrained by IFC version fragmentation (IFC2x3, IFC4, IFC4x3), tool-dependent PropertySet usage (e.g., Revit vs. Archicad exports), and the limited formalisation of spatial/topological predicates required for domains such as fire safety and egress. Prior research shows that even formally aligned IFCOWL models may fail when geometric or relational constraints expose schema gaps or incomplete modelling practices [7,36]. Consequently, semantic alignment should be understood not only as a technical RDF mapping problem, but as a socio-technical interoperability challenge shaped by modelling conventions and execution plans.

Across these studies, the most complete alignment relies on IFCOWL with RDF graphs and SWRL, SHACL, or SPARQL logic. These methods connect rule meaning directly to IFC entities, PropertySets, and spatial relations, enabling traceable queries and explicit links between each regulation clause and its related model element. However, their robustness depends on consistent IFC schema versions and modelling practices across authoring tools, as fragmentation and heterogeneous PropertySet usage can weaken alignment reliability. Ongoing challenges therefore include maintaining cross-version compatibility, addressing schema gaps in spatial and relational predicates, handling rules that depend on time or context, and establishing shared benchmarks to assess the precision and transferability of rule-to-IFC mappings. To make these interoperability differences more explicit, Table 5 summarises representative semantic-alignment approaches and highlights how version dependence, PropertySet conventions, and tool-specific modelling practices affect portability across ACC workflows.

As Table 5 shows, approaches grounded in IFCOWL and standards-based constraint languages generally offer stronger traceability, whereas tool-bound implementations remain more exposed to export conventions and PropertySet variability. Overall, this stage is most robust when rule-to-model bindings are explicit and standards-based, but it remains vulnerable where IFC-version differences, PropertySet variability, and tool-specific modelling practices weaken portability and traceability.

5.2.2. *Integration of ontologies and knowledge graphs*

Ontology and knowledge graph-based pipelines formalise regulatory concepts, connect them with BIM and IFC data, and support semantic reasoning for automated compliance. Multi-ontology fusion integrates regulatory, design, and checking ontologies through mapping rules, forming a shared knowledge base where SPARQL, OWL, and SWRL rules operate on IFC-derived models [3]. Tool-based frameworks extend this workflow by converting IFC data into IFCOWL and RDF and using engines such as Apache Jena to validate models against formal specifications [79]. These tools demonstrate vendor-neutral and explainable model checking within shared data environments.

Platform-scale efforts build complete compliance pipelines that integrate NLP and semantic web components [80]. These pipelines curate national regulation corpora, construct knowledge graphs, and provide semantic search and enrichment services that support rule authoring and checking workflows. At the same time, research on the semantic enrichment of BIM applies hybrid ML to classify spatial entities, while rule-based inference composes higher-order constructs, such as tenements [81]. This integration supplies the ontological detail that ACC requires for accurate rule execution.

Ontology-based compliance checking for structural codes illustrates the approach used and the foundation of this work. In this approach, IFCOWL works with SWRL or Drools to define variables, conditions, and rule flows, allowing direct verification of design data [82]. Related studies utilise ontology-based semantic profiling of regulations, incorporating SKOS (Simple Knowledge Organisation System) terms, deontic logic, and clause structures, to enhance traceability and ensure consistent rule interpretation [83]. These refinements strengthen the alignment of knowledge graphs and increase the reliability of subsequent reasoning.

Persistent gaps remain. Cross-jurisdictional ontology differences limit interoperability, and most systems still depend on manual mapping between regulation and design concepts. Benchmarks for ontology and knowledge-graph alignment, as well as rule coverage, remain scarce, and hybrid symbolic ML systems lack standardised provenance records for audit purposes. In addition, during knowledge-graph construction, rich clause semantics (e.g., applicability, exceptions, and measurement context) can be compressed into simplified nodes and relations, which may reduce retrieval fidelity and downstream checking accuracy if not controlled. Progress in this area requires shared IFCOWL-linked corpora, reusable mapping patterns for PropertySets

and topology, and integrated OWL, SWRL, and SPARQL evaluation suites that measure fidelity from regulatory clauses to model elements across domains.

5.3. Rule execution

Rule execution concerns the ACC tasks through which formalised requirements are operationalised and applied to prepared model data to generate compliance outcomes. Within this stage, Rule Representation and Reasoning and Model–Driven Compliance Intelligence provide the main analytical lenses. The core tasks include encoding executable rules, binding them to model content, resolving applicability conditions, and running checks through symbolic, semantic, or hybrid execution mechanisms. Representation choices include logic clauses, rule trees, SHACL, SWRL, SPARQL, JSON rule frames, and ontology-grounded execution artefacts, while the dominant techniques span symbolic reasoning, knowledge-graph-supported inference, retrieval-grounded LLM pipelines, and hybrid neural–symbolic execution. The main meta-properties at this stage are transparency of execution, fidelity to source clauses, grounding reliability, and the extent to which outcomes remain auditable and reproducible in practice.

5.3.1. Rule representation and reasoning

Rule representation and reasoning in ACC aims to convert regulatory requirements into logic-based, computable forms that allow transparent and machine-executable compliance. The main effort lies in encoding natural language clauses as formal logic statements or semantic rules that reasoning engines can process. Early systems used logic programming and invariant reasoning to verify model properties and assess compliance in a structured and interpretable way [84].

Later studies extended earlier methods by expanding the rule set to improve scalability. Iterative refinement is used to generate refined clauses, increasing precision and coverage while maintaining clear links to the original text [85]. Hybrid reasoning combined symbolic and data-driven logic to manage nested clauses [86]. Knowledge graphs captured relationships between design constraints, improving semantic coverage and rule reuse [86].

As AI and NLP have advanced, recent frameworks have begun using LLMs to assist in rule translation, thereby reducing the cost of manual rule writing without compromising logical consistency or semantic clarity [87]. Broader semantic web systems now integrate rule reasoning with formal ontologies and simulation data, allowing dynamic validation across both regulatory and simulation environments [88].

Persistent challenges include the absence of standardised rule schemas and insufficient interoperability among systems designed to process performance-based or multimodal regulations. Additionally, the unavailability of shared datasets restricts the objective assessment of rule precision and reasoning quality. Symbolic systems, which rely on explicit logical rules and formal representations, are vulnerable when processing ambiguous or incomplete data. In contrast, neural models learn patterns from data but frequently lack transparency in their decision-making processes. Overall, this line of work is comparatively mature in symbolic and ontology-grounded execution, but remains less stable when representations become heterogeneous or when neural assistance weakens traceability to source clauses.

5.3.2. Model–driven compliance intelligence

ACC has evolved from handcrafted rules toward learning-driven pipelines that link LLMs, deep neural networks, knowledge graphs, and ontological grounding. Hybrid frameworks now combine language models with pre-classifiers and domain ontologies to reduce annotation effort and enhance the extraction of structured rules from complex regulatory clauses [28]. These frameworks show measurable gains in end-to-end automated code checking. A complementary line of work fuses knowledge graphs with LLMs [89]. Standards-based graphs guide model reasoning through prompt templates and schema-aware retrieval, improving accuracy in construction-scheme validation. While this grounding can improve traceability, the conversion of regulatory text and BIM semantics into graph structures can introduce information loss or semantic simplification. Nuances such as exceptions, conditional scope, cross-references, and unit/measurement context may be omitted or normalised, so retrieval can return incomplete evidence and the model may produce overly general rules. This can degrade system performance by increasing false positives/negatives (missing constraints) and causing brittle behaviour on long or highly-referenced clauses. A practical mitigation is to retain clause-level provenance links from knowledge graph facts back to the original clause spans and IFC elements, and to keep the retrieved clause text available for verification before executing checks. In practice, these prompt templates commonly follow a few recurring patterns for clause interpretation and clause-to-rule translation: stepwise (decomposed) prompting, explicit slot-filling to structure clauses into slots; few-shot exemplars to stabilise outputs; schema-constrained outputs (e.g., JSON rule frames) for downstream parsing; and retrieval-grounded verification instructions to reduce unit/threshold errors and hallucinated constraints.

At the representation layer, transformer-based alignment connects IFC entities with regulatory concepts through relation classification [11]. The process uses IFC definitions and an IFC knowledge graph to close the semantic gap between regulatory text and building model data. This alignment supports consistent downstream checking. In parallel, deep learning pipelines extract relations from safety requirements and translate them into graph queries [90]. These pipelines decompose long or nested prescriptions into executable structures suitable for automated reasoning.

Domain-specific frameworks that pair pretrained language models with ontologies extend this trend. Studies combine entity and relation extraction with spatial–semantic trees and ontology mapping to build logical clauses with high formal accuracy in domains such as fire safety [91]. Exploratory prompt-based research compares outputs from language models with advanced semantic rule pipelines, showing faster translation from text to logic rules but continued limits in precision and consistency [92].

However, evaluations under realistic regulatory language show that LLM-assisted rule generation can still degrade when specialised constraints, nested prescriptions, or implicit cross-references materially affect compliance outcomes. Empirical comparisons in building-code transformation demonstrate that LLMs can reduce authoring effort, but still fall short of semantic rule-based conversion when handling specialised constraints [92]. Knowledge-guided prompting further shows that weak control of grounding and prompts can produce excessive checking points, misaligned standard rules, and unit or threshold misinterpretations, which can propagate into downstream checks [89]. In ACC, these failure modes are particularly consequential because unsupported constraints or unit/threshold misinterpretations may be compiled into executable checks, leading to incorrect compliance verdicts and elevated responsibility risks in practice. Risk control therefore focuses on constraining and validating LLM outputs through retrieval-grounded prompting with evidence links back to the source clause [93], schema- and unit-constrained rule frames [69], automated consistency checks, and downstream symbolic verification against ontologies and rule engines, alongside mandatory expert review supported by end-to-end provenance capture [20,94]. Accordingly, the literature points toward using LLMs primarily as decision-support within hybrid pipelines, where retrieval grounding [93], schema-aware outputs and explicit unit/threshold normalisation, combined with downstream symbolic validation, provide assurance, transparency, and accountability in compliance workflows [20,94]. To make these practitioner-relevant risks more explicit, Table 6 summarises the main LLM failure modes reported in recent ACC studies on regulatory interpretation, together with their likely effects on compliance outcomes and the safeguards most often proposed in the literature.

Table 6
Brief summary of LLM failure modes in regulatory interpretation for ACC.

Failure mode	ACC implication	Typical safeguard
Sensitivity to domain-specific constraints	Specialised regulatory meanings may be oversimplified or misinterpreted during rule generation.	Domain-grounded retrieval, ontology support, and expert review.
Unit and threshold misinterpretation	Numeric limits or comparison conditions may be translated incorrectly into executable checks.	Unit-constrained rule frames, normalisation, and consistency checks.
Misaligned or excessive checking outputs	Generated checks may not align closely with the source clause or may introduce unnecessary checking points.	Retrieval-grounded prompting, clause-level evidence links, and manual validation.
Implicit cross-reference failure	Dependencies, exceptions, or referenced clauses may be missed during interpretation.	Structured clause decomposition, provenance links, and human verification.
Opaque reasoning and unsupported outputs	Practitioners may be unable to verify how a rule was produced or whether it is fully supported by the source text.	Provenance capture, symbolic verification, and reviewer-facing explanations.

Across these studies, the main technical task is no longer simple rule encoding but reliable retrieval-grounded support for rule interpretation and checking under strict provenance and validation constraints. Models need a traceable reasoning path that links clause interpretation to rule formation and compliance checks. Shared benchmarks are missing for cross-jurisdiction evaluation, rule coverage, and robustness testing. Retrieval and grounding methods must better unify the knowledge graph and ontology context with BIM data during inference. Progress will depend on hybrid neural–symbolic systems that integrate prompts or controllers linked to ontologies, shared corpora binding regulations to IFC data, and evaluation frameworks that score both reasoning quality and field reliability.

5.4. Reporting

At the reporting stage, studies differ mainly in how they communicate compliance outcomes, connect them to source requirements and model elements, and support inspection. Reporting formats include textual and visual outputs, as well as standards-based reports that link violations to the relevant rules and model elements [20,95]. Other approaches move beyond simple pass/fail outcomes by assigning graded compliance scores or generating problem summaries and corrective suggestions [33,35]. Reporting also varies in its support for different standards and integration with BIM environments [7,96]. Traceability and reviewability are supported by distinguishing checking conditions from requirements and keeping rule classifications open to expert inspection [94,97]. Object-centred reports can further explain why individual checks fail [98]. Despite these advances, reporting remains fragmented across tools and standards. No shared output structure consistently links verdicts to source clauses, executed rules, affected model elements, and supporting evidence, while few studies evaluate whether practitioners can understand and act on these reports. Reporting therefore remains insufficiently standardised and empirically validated for cross-platform decision support.

5.5. Decision support

Decision support concerns how reported compliance outcomes are used in practice to assist review, corrective action, redesign, and workflow integration. Within this stage, Tool Development and Real-World Application provides the principal mapped theme. The core tasks include embedding checking outputs into design and permitting workflows, enabling practitioner interaction with compliance results, and supporting action on identified violations or recommendations. Representation choices include prototype platforms, BIM-integrated interfaces, workflow tools, compliance dashboards, and case-based reporting environments. The dominant techniques involve ontology-driven compliance tools, BIM plug-ins, linked-data platforms, rule-authoring environments, and deployment-oriented validation studies. The main meta-properties at this stage are usability, workflow fit, adoption readiness, interoperability with practice environments, and the degree to which systems provide actionable support rather than isolated technical outputs.

5.5.1. Tool development and real-world application

Recent advances in ACC now centre on tools applied directly within real construction projects. These projects translate research models into practical tools that integrate BIM, ontologies, and regulatory reasoning. The focus has shifted from abstract rule design to evaluating how well ACC tools perform, scale, and hold up under real-world conditions.

SmartNorms4BIM was the first complete prototype to combine OWL2-based ontological reasoning with IFC models for code compliance [95]. The prototype performs semantic classification of building elements and generates both textual and visual reports. It was applied to a housing regulation case, showing how ontology-driven reasoning can operate within real BIM projects.

Another research presents tool-level integration through a domain-specific language built for regulatory encoding and BIM-based verification [96]. It operates within standard design platforms, allowing users to create rules, automate checks, and confirm results through human review. This level of integration moves compliance checking closer to regular use in architectural and engineering work.

A related framework links NLP, ontology construction, and BIM data to support design-for-safety checks in real-world projects [99]. A hospital case study tested the framework, demonstrating high precision and recall in automated rule extraction and verification, which confirms that NLP-based ACC systems can function reliably in practice.

Another study focuses on system validation, evaluating interoperability and performance through targeted pilot testing [20]. The findings underline the need for standardised rule formats and user-centred interfaces to ensure that compliance tools align with professional design practices.

The BRISE-Plandok project provides the data foundation by developing an annotated regulatory corpus and baseline models that automate rule formalisation [100]. The I-REC project initiative builds on this foundation with a linked-data knowledge graph that links regulatory terms to construction vocabularies [101]. Both resources strengthen the ecosystem that supports practical ACC applications.

Together, these efforts show a clear shift from conceptual research to deployed, ontology and data-driven systems that deliver regulatory reasoning, workflow automation, and human oversight. Challenges in scalability, data exchange, and interface design remain, yet the new generation of tools demonstrates that ACC is moving steadily from prototype to industry practice.

5.6. Assurance and validation

Assurance and validation are treated in this review as cross-cutting analytical concerns spanning all ACC stages rather than as a separate pipeline function. The main tasks considered here involve explaining compliance outcomes, enabling expert review and intervention, and evaluating system performance through reproducible validation and benchmarking. Representation choices include provenance records, explanation artefacts, review interfaces, evaluation datasets, and benchmark structures, while the dominant techniques include standards-based validation, human-in-the-loop workflows, comparative benchmarking, and user-oriented assessment. Across these studies, the key meta-properties are trust, accountability, reproducibility, robustness, and readiness for real-world deployment.

5.6.1. Explainability and trust in AI systems

Explainability in ACC is a core design requirement that defines how the data model, rule language, and interface are structured. It is essential to separate technical traceability from user-centred explainability. Traceability concerns whether a compliance system exposes inspectable artefacts, such as formalised rules, execution traces, and provenance links, so that the steps leading to a verdict can be reconstructed and verified. User-centred explainability addresses a different question: whether practitioners, including architects, inspectors, and auditors, can understand the rationale presented, relate it to their tasks, and make informed decisions on that basis. Across the reviewed corpus, empirical validation of stakeholder comprehension and actionability is reported inconsistently. Research shows that symbolic and standards-based frameworks, including RDF graphs combined with OWL, SWRL, SPARQL, or SHACL constraints, provide a transparent and technically auditable foundation for compliance verification [7]. These formats expose both inputs and validation outcomes in interoperable and queryable forms, creating traceable links between a clause, its constraint, the executed check, and the resulting verdict.

Operational systems show how this works in practice. A guardrail inspection framework encodes safety logic as declarative rules. It prioritises transparency, completeness assessment, extensibility, and what it terms ‘‘pessimism’’, meaning a violation is flagged unless it can be ruled out [94]. The pipeline remains open for review, allowing each classification to be examined and corrected by site staff as needed. This white-box process is designed to support practitioner trust because the reasoning steps are explicit and can be inspected and queried when disagreements arise.

A related object-centred method builds checks around BIM objects and their attributes. It combines textual rule checking with tests of model completeness, intelligent object behaviour, and design compliance [98]. The approach not only increases checking speed but also pinpoints the exact place in the object hierarchy where a failure occurs. Moreover, it improves the clarity of explanations and strengthens confidence in the system’s recommendations.

LLM-based workflows open new possibilities but also raise questions of trust. One framework converts regulatory clauses into Python checks within a BIM environment, providing wide coverage and rapid results [35]. Yet it exposes the need for provenance: prompts, retrieved context, generated code, and model evidence must be stored so reviewers can see why a space failed and how a clause was applied.

An experimental autonomous agent takes it a step further by planning, retrieving BIM data, and running checks without direct supervision [102]. The researchers emphasise that simple pass-fail reports are insufficient. Transparent reports and human oversight are necessary when rules are ambiguous or involve multiple data types, as these controls help maintain confidence in the results.

Across the literature, four shared needs emerge. First, unified provenance that links clause text, rule files, retrieval context, generated code, BIM element references, and evidence snapshots in a single, machine-readable trail. Second, role-aware explanations that vary in depth for inspectors, designers, and regulators, but draw from a single authoritative source (for example, BIM-linked visual logic trees and element highlights for designers, and plain-language justifications with clause citations for regulators). Third, hybrid transparency for neural components so prompts, retrieved passages, and code revisions are recorded and reviewable. Fourth, evaluation metrics that measure not only accuracy but also explanation quality, usefulness, and trust over time. However, few studies report controlled evaluations of stakeholder comprehension or decision impact, leaving a gap between technical transparency and demonstrated human-centred explainability. Together, these aims define compliance systems where every inference can be seen, challenged, and corrected. This provides a foundation for trustworthy ACC, but the field still lacks consistent evidence that technically traceable systems also produce explanations that stakeholders can reliably understand and act upon.

5.6.2. Human-in-the-loop approaches

Recent developments in automated verification compliance underscore the importance of human oversight in AI-driven workflows. HITL frameworks can also directly supervise the compliance pipeline to make the generation of rules, interpretation, and verification transparent and auditable, ensuring consistency from the perspective of professional judgment. These systems recognise that interpreting regulations requires a contextual understanding and logical reasoning. While automation enhances speed and efficiency, human judgment remains an essential component in the process.

During data preparation, the selection of high-quality, manually labelled datasets has become essential for practical model training. These structured datasets are designed for rule extraction and help domain experts label, verify, and refine machine-generated results [12]. This process generates feedback loops that improve both accuracy and contextual relevance. Building on this idea, adaptive learning pipelines merge LLMs with active human validation to refine the semantic tagging of regulatory clauses. In these processes, experts correct any misclassifications made by the model, gradually teaching the model to generate more understandable regulatory structures [29].

HITL principles also guide the reasoning and validation stages of compliance checking. Standards-based validation frameworks that leverage constraint languages such as SHACL allow users to inspect, question, and revise automated reasoning findings while maintaining a human review layer over machine inference [103]. Infrastructure rule creation and design verification now connect structured engineering knowledge with clear modelling interfaces. These tools let domain experts follow and refine rule logic at every stage of the checking process. Participatory workflows are particularly common in systems that use graph-based or object-oriented modelling, where manual input defines the ontology, rule constraints, and test cases [104]. Semi-automated learning systems follow a similar pattern. They use graph-based models for rule induction but keep a supervised verification step, allowing human reviewers to assess accuracy and clarity before deployment [105].

These studies show that effective compliance automation depends not only on model performance but also on structured, transparent collaboration between humans and machines. Human-in-the-loop frameworks embed expert review and feedback throughout data annotation, rule creation, and validation. However, existing approaches still lack standardised benchmarks for measuring the quality and consistency of human interventions across different regulatory domains. There is also limited research on how HITL pipelines scale when regulatory updates occur rapidly or simultaneously across multiple jurisdictions.

5.6.3. Evaluation and benchmarking

Evaluation in ACC spans system-level benchmarking and task-specific testing for rule execution and information extraction. Studies that focus on RDF-native compliance treat benchmarking as a balance between reasoning efficiency and data fidelity [106]. Comparisons

between SHACL and ASP (Answer Set Programming)/DLV2 show that avoiding data conversions improves scalability and stability. One study formalises this through a multi-reasoner benchmark using simulated datasets of growing size, offering a rare baseline for performance in legal and compliance contexts [106].

Task-driven evaluations measure accuracy, precision, recall, and F1-score, alongside the quality of end-to-end checking. For clarity, this review adopts the conventional confusion-matrix definitions for binary classification (and for extraction tasks once $TP/FP/FN/TN$ are defined at the chosen unit of analysis): precision $P = \frac{TP}{TP+FP}$, recall $R = \frac{TP}{TP+FN}$, $F1 = \frac{2PR}{P+R}$, and accuracy $Acc = \frac{TP+TN}{TP+TN+FP+FN}$. Graded (partial) compliance scoring denotes an ordinal or continuous measure of satisfaction across multiple sub-constraints rather than a binary pass/fail outcome; since studies operationalise these metrics at different units (token/span/clause/rule level), results are reported as defined by the original authors and direct comparisons are noted where limited. In a concrete dam construction use case, a model that combines entity extraction, syntax-tree parsing, and semantic alignment reported an entity extraction F1 improvement of 12.89% and 88.89% checking accuracy [33]. It also introduced a scoring-based checking outcome to reflect partial compliance rather than simple pass/fail outcomes.

Recent methodological research defines what counts as an adequate report. One study on semantic code representation proposes that results should separate applicability from requirement, and integrity from violation [97]. It also requires traceable mappings to IFC or IFCOWL to enhance both usability and interpretability. These distinctions serve as practical rubrics for assessing report completeness, rule coverage, and traceability quality.

Model-driven research extends evaluation to LLM pipelines. A GPT-based framework tests prompt design, model stability, and text-to-rule reasoning, treating prompt engineering as an explicit evaluation target [107]. It recommends structured test sources that assess sensitivity and reasoning validity in addition to classic effectiveness metrics.

System-level studies in related domains provide further guidance for benchmarking. A knowledge-informed ACC pipeline with Performance-Based Generative Design (PGD) integration reduced runtime by 73.25% by using knowledge graphs to prune design spaces and support cross-domain reasoning [108]. Although not a direct ACC benchmark, it illustrates how efficiency and solution quality can influence deployment readiness.

Current gaps are evident. At present, the evidence base for benchmarking is distributed across partially overlapping resources rather than a single integrated suite. For example, RDF-native studies provide scalable, multi-reasoner performance baselines on simulated datasets [106], while ACC-oriented research contributes regulatory corpora, semantic parsing artefacts, and formalisation pipelines for transforming building codes into machine-interpretable representations [72,73]. Related work on IFC-linked validation and cross-method comparison further establishes reproducible datasets and rule-to-model mapping approaches within specific domains [7]. Task-driven evaluations introduce graded outcome scoring that better reflects partial compliance than binary judgments [33]. However, these components are rarely combined into shared challenge datasets that jointly package regulatory text, annotated logical forms, IFC instances, and expert-validated ground-truth compliance outcomes, which limits reproducibility and cross-study comparability. Comparative evaluations across jurisdictions remain limited, and few studies present dashboards that track multiple metrics, including precision, recall, F1-score, rule coverage, runtime, memory usage, and usability. Neural and LLM-based systems also require metrics that account for explainability, including provenance completeness and rationale faithfulness, as well as stress tests that examine prompt robustness and generalisation. Future benchmarking efforts should therefore converge on a standardised case structure ⟨Clause, LogicalForm, IFCModel, Outcome⟩ with fixed splits and baseline runs, unify shared reasoning tasks, employ graded scoring, and establish clear reporting standards [97,107]. Such efforts should also publish open IFC-linked datasets, adopt uniform evaluation suites, measure human factors such as learnability and error recovery to gauge system maturity and readiness for real-world use, and track community uptake of shared benchmark formats over time to assess whether these standards achieve practical adoption.

6. Discussion and future directions

The stage-based synthesis reveals both substantial technical progress and persistent limitations across ACC research. These findings are considered in relation to the research questions before the principal implications and future research priorities are presented.

6.1. Overview and synthesis

The reviewed studies show clear but uneven progress in ACC. The strongest advances appear in structured rule formalisation, ontology- and IFC-linked semantic alignment, and executable reasoning pipelines that translate regulatory text into computable logic. These developments have strengthened the technical basis for auditable compliance workflows and have moved parts of ACC beyond isolated proof-of-concept implementations. At the same time, multimodal regulatory interpretation, stakeholder-facing explainability, and end-to-end validation across heterogeneous models and jurisdictions remain less mature and, in some cases, unresolved at field level. As a result, recent ACC research is better characterised by selective methodological consolidation than by uniform maturity across the pipeline. More specifically, the reviewed evidence indicates comparative maturity in text-based regulatory parsing, rule formalisation, ontology- and IFC-linked semantic alignment in controlled settings, and symbolic rule execution. By contrast, robust multimodal interpretation, cross-version and cross-tool interoperability, verifiable LLM-assisted reasoning, stakeholder-facing explainability, and shared end-to-end benchmark datasets remain unresolved at field level.

The literature presents consistent technical growth. Early work on regulatory text interpretation demonstrates that grammar-based and hierarchical parsers can structure complex clauses and transform them into semantic computable requirements [69,70]. Intermediate representations further improve semantic parsing into machine-readable rules [72]. Progress in semantic alignment with BIM and IFC data has powered this base. Ontology-based systems using ifcOWL, SPARQL, OWL, and SWRL support direct checking and reasoning over IFC models [74,75]. OntoBPR extends this approach with SHACL-based building-permit checking using standardised information containers [76]. High-level implementable methods and I-SNACC further connect structured rules with automated BIM checking [67,68]. Knowledge-driven pipelines combine multi-ontology BIM reasoning with vendor-neutral IFC validation, enabling explainable verification [3,79]. SHACL validation and technical specifications support inspectable and deployable ACC tools [103,104], while human-in-the-loop supervision keeps reasoning transparent and auditable [94]. At the same time, integrating LLMs into ACC introduces regulatory risks. Initial testing identified limitations concerning domain-specific concepts and units, whereas knowledge-graph grounding helps align LLM-based checking with domain rules [89,92]. These risks motivate verifiable hybrid pipelines. Knowledge-graph representation and ontology-enhanced NLP strengthen regulatory grounding [90,91], while SHACL validation and expert-controlled inspection support symbolic checking, transparent reporting, and human oversight [20,94]. Recent work is increasingly focused on benchmarking and deployment. BIM-linked prototypes support rule checking, validation, and visual reporting [20,95], while BRISE-Plandok provides an annotated legal corpus for evaluating regulatory rule extraction [100].

However, three technical bottlenecks still limit broader field-level maturity. First, few systems integrate text, tables, diagrams, and operational data within a single grounded compliance pipeline. Second,

semantic interoperability remains constrained by IFC-version fragmentation, heterogeneous PropertySet practices, and incomplete mappings between regulatory meaning and model semantics. Third, many AI-assisted approaches reduce manual effort in rule interpretation and drafting, but they still depend on symbolic checking, provenance capture, and expert review to provide verifiable compliance outcomes. In summary, current ACC progress is strongest in structured, ontology-grounded, and domain-bounded automation, whereas robust multimodal, interoperable, benchmarked, and fully trustworthy end-to-end automation remains unresolved.

AI and LLMs are contributing to several ACC tasks, but the reviewed evidence indicates that their most reliable role remains supportive rather than fully autonomous. Transformer and vision-language models are widening extraction beyond plain text, and LLMs can assist clause structuring, candidate rule drafting, and semantic matching to BIM/IFC concepts. Their value is greatest when combined with ontology grounding, retrieval support, symbolic validation, provenance capture, and expert review. Current progress therefore points toward hybrid symbolic–neural pipelines, not toward standalone LLM-based compliance engines.

6.2. Linking the stage-based synthesis to the research questions

RQ1. The review indicates that ACC has reached comparative methodological maturity in three areas: (i) text-based regulatory parsing and clause decomposition, (ii) formalisation of rules into logic-ready representations, and (iii) ontology- or IFC-linked execution pipelines for well-structured and domain-bounded checks. In this sense, the recent methodological shift is not that LLMs have solved ACC end-to-end, but that they have reduced manual effort in upstream tasks such as clause tagging, rule drafting, and semantic alignment support. The risks associated with standalone LLM-based compliance checking therefore motivate the development of hybrid pipelines, in which LLMs serve as assistants alongside retrieval, ontologies, or downstream symbolic validation. By contrast, multimodal regulatory interpretation, robust cross-document reasoning, and fully trustworthy end-to-end automation remain unresolved. Evaluation has also improved, but mostly through isolated case studies, task-specific metrics, and prototype-level validation rather than shared, cumulative benchmark practice. Therefore, the main methodological development since 2022 is the rise of hybrid symbolic–neural ACC pipelines, while the main unresolved issue is the lack of a reproducible end-to-end validation culture across jurisdictions, model types, and rule complexities.

RQ2. The reviewed evidence shows that the mapped technical themes do not manifest evenly across the ACC stages. In the upstream stages of rule interpretation, the field shows stronger maturity in text-centred clause parsing and rule formalisation than in genuinely multimodal interpretation across text, tables, figures, and layout dependencies. In model preparation, ontology-backed BIM/IFC alignment has become more structured and technically explicit, but interoperability remains constrained by IFC-version fragmentation, heterogeneous PropertySet practices, and tool-dependent modelling conventions. In rule execution, symbolic and ontology-grounded reasoning are comparatively more mature than LLM-assisted execution, whose most reliable role remains supportive within hybrid pipelines rather than fully autonomous. In the downstream stages, reporting and decision support are becoming more visible, but they remain less standardised in terms of traceability, practitioner-facing usability, and workflow integration. Overall, the dominant technical direction is therefore not a replacement of symbolic ACC by LLMs, but a selective hybridisation in which mapped themes manifest differently across stages depending on the maturity of their tasks, representations, techniques, and assurance requirements.

RQ3. Three bottlenecks remain unresolved and currently prevent field-level robustness. First, interoperability is still constrained by fragmented IFC versions, tool-dependent PropertySet practices, inconsistent modelling conventions, and the absence of stable mappings between regulatory semantics and model semantics. This means that many ACC solutions remain portable only within narrow technical settings. Second, explainability is still stronger at the level of traceable rule execution than at the level of stakeholder-facing justification. Many systems can expose rules, queries, or validation traces, but far fewer demonstrate that architects, inspectors, or regulators can meaningfully understand and act on those explanations. Third, validation remains fragmented because few studies evaluate ACC systems against shared datasets that jointly combine regulatory clauses, logical forms, BIM/IFC instances, and expert-validated outcomes. Recent works have started to address these problems through ontology-linked pipelines, SHACL- and RDF-based validation, human-in-the-loop review, provenance-aware workflows, and benchmark proposals. However, none of these issues can yet be regarded as solved at field level. The central boundary of current progress is therefore clear: ACC is increasingly capable of producing structured, executable, and partially explainable checks in controlled settings, but it still lacks interoperable, benchmarked, and stakeholder-validated end-to-end robustness for broad deployment.

6.3. Key findings

Building on the stage-based synthesis presented in Section 5, the following findings capture the core scientific outcomes of the review:

- **Multimodal extraction lags behind BIM/IFC alignment** (Sections 5.1–5.2). Text-centred parsing and IFC-linked semantic alignment have advanced substantially, but robust multimodal interpretation across text, tables, diagrams, and cross-references is still not achieved at field level. This remains one of the clearest unresolved bottlenecks in current ACC research.
- **Rule formalisation and reasoning are comparatively mature** (Sections 5.1–5.3). Recent studies show comparatively mature progress in transforming regulations into logic-ready, ontology-linked representations and executing them through symbolic reasoning frameworks. However, reasoning becomes less consistent when pipelines depend on heterogeneous representations or weakly grounded neural generation.
- **Knowledge-driven AI is shifting ACC toward hybrid reasoning** (Section 5.3). Ontologies and knowledge graphs are increasingly being combined with LLMs and deep learning to reduce manual rule authoring and improve adaptability. Their most reliable role, however, remains assistive and grounded within hybrid pipelines rather than fully autonomous compliance decision-making.
- **Trust, reporting quality, and human oversight remain essential** (Sections 5.4–5.6). Reporting, explainability, and human-in-the-loop workflows are no longer optional add-ons but necessary conditions for trustworthy ACC. Yet they remain insufficiently systematised, with limited evidence on reporting completeness, stakeholder-facing explanation quality, user trust, and decision-support effectiveness.
- **Evaluation and decision-support remain fragmented** (Sections 5.5 and 5.6). Benchmarking and practical decision-support deployment are increasing, but evaluation is still too fragmented to support cumulative comparison across methods, datasets, and jurisdictions. The absence of shared benchmark structures continues to limit reproducibility, scalability, and broader deployment confidence.

6.4. Future directions

The future directions below are motivated by two consistent patterns in the stage-based synthesis (Sections 5.1–5.6): variation in emphasis and evidence coverage across ACC pipeline stages, and limited

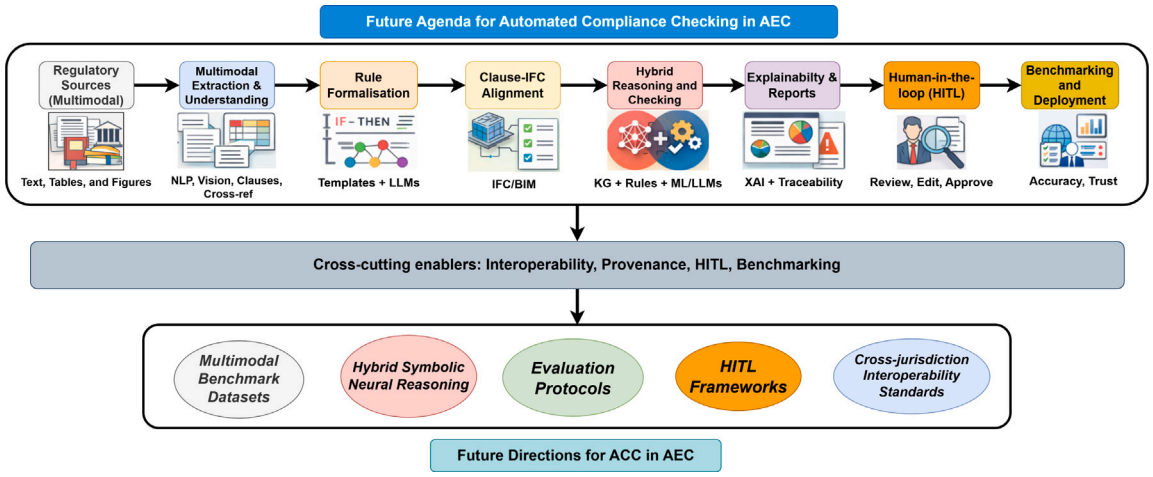


Fig. 9. Conceptual research agenda for ACC, situating the proposed future directions within the stages and cross-cutting enablers of an end-to-end compliance pipeline.

cross-study comparability due to fragmented datasets and inconsistent benchmarking. Fig. 9 summarises the end-to-end pipeline and maps the proposed directions to the stages and cross-cutting enablers they target, including interoperability, provenance, reporting quality, human-in-the-loop oversight, and benchmarking.

Open, ontology-linked multimodal benchmark datasets. Future work should prioritise open benchmark datasets that bundle clause-segmented regulatory sources with cross-references, table and figure region annotations with extracted structured content, gold rule representations, and IFC or BIM linked ground-truth compliance labels. Benchmark packaging should align with semantic web and rule formalisms already used in ACC, such as RDF or IFCOWL, SHACL or SWRL, and LegalRuleML, and should explicitly document what remains uncovered end-to-end. Current evaluation remains fragmented, and few shared datasets jointly bind clauses, representations, IFC instances, and compliance outcomes into a reproducible suite. Realising this direction requires coordinated community curation, governance for licensing and access, and shared annotation guidelines that remain useable across jurisdictions and toolchains. Future work should also monitor community adoption of such benchmark structures, for example by tracking reuse across datasets, shared tasks, baseline implementations, and cross-jurisdiction evaluation studies.

Hybrid symbolic neural reasoning with explainable inference. Research should develop hybrid symbolic neural reasoning engines that provide explainable inference paths. Hybridisation is accelerating, but neural components remain difficult to verify without explicit provenance, controllable retrieval, and auditable reasoning traces. Progress here requires integrating retrieval grounding and LLM outputs with logic-based checking and trace capture, supported by industry validation of scalability and workflow fit within real BIM environments.

Evaluation protocols that combine technical and human factors. A further priority is to establish evaluation protocols that combine technical performance metrics with measures of human trust and usability. Benchmarking is increasing, yet evaluation remains uneven, and user-centred validation of explainability and actionability is limited. Addressing this requires agreed metric sets for accuracy and robustness alongside human-focused measures, such as comprehension, workload, and decision support value, supported by consistent reporting templates.

Participatory human in the loop frameworks. Future systems should embed expert feedback throughout AI-assisted compliance checking using participatory human in the loop frameworks. Human oversight is widely recognised as essential, but it is not yet systematised into repeatable workflows across the pipeline. Delivering this requires co-design with practitioners, including designers, inspectors, and auditors, clear accountability roles, and feedback loops that update rules, prompts, and ontologies without eroding traceability.

Cross-jurisdiction interoperability standards. Broader uptake depends on cross-jurisdiction interoperability standards that align rule ontologies and BIM schemas at scale. Interoperability barriers persist due to semantic gaps between ontologies and evolving IFC versions and modelling practices. Advancing this direction requires joint research and standards work to harmonise rule vocabularies with BIM schemas, alongside industry adoption support through tool vendors and engagement with regulators.

7. Conclusion

This paper reviewed ACC research in AEC published between 2022 and 2025 using an AI-assisted PRISMA workflow organised around five ACC stages and using mapped technical themes as analytical lenses. Together, these themes comprise Multimodal Information Extraction, Formalisation of Regulatory Text, Semantic Alignment with BIM/IFC, Integration of Ontologies and Knowledge Graphs, Rule Representation and Reasoning, Model-Driven Compliance Intelligence, Tool Development and Real-World Application, Explainability and Trust in AI Systems, Human-in-the-Loop Approaches, and Evaluation and Benchmarking. The review provides a transparent and repeatable evidence-synthesis process that combines structured screening, thematic mapping, and manual validation to capture recent methodological shifts in AI- and LLM-enabled ACC.

The synthesis shows that recent ACC progress is real but uneven. The strongest signs of methodological consolidation appear in rule formalisation, semantic alignment with BIM/IFC, ontology- and knowledge-graph integration, rule representation and reasoning, and tool development in real-world settings. These areas now provide a comparatively stable foundation for structured and auditable compliance workflows. By contrast, multimodal extraction, model-driven

compliance intelligence, and stakeholder-facing explainability remain less mature, while human-in-the-loop practice and benchmarking are advancing but still fragmented across studies. Across the field as a whole, interoperability remains limited, rule authoring still depends substantially on expert effort, and the scarcity of open benchmark datasets continues to restrict fair comparison and reproducibility.

The review shows that ACC research reported between 2022 and 2025 is increasingly characterised by hybrid reasoning architectures that combine semantic web technologies with AI- and LLM-enabled components. Interpreted against the symbolic and ontology-based foundations established in earlier reviews, this recent literature indicates a clear shift toward more adaptive and knowledge-driven compliance workflows (RQ1).

Current ACC research also shows uneven but recognisable technical maturity across the reviewed themes. Rule formalisation, semantic alignment with BIM/IFC, ontology- and knowledge-graph integration, rule representation and reasoning, and tool development in real-world settings are comparatively more mature. By contrast, multimodal extraction, model-driven compliance intelligence, and stakeholder-facing explainability remain less mature, while human-in-the-loop practice and benchmarking are progressing but still fragmented. Overall, the dominant technical direction is hybridisation, in which semantic-web and IFC-grounded approaches are increasingly combined with AI- and LLM-enabled components, but not yet with uniform maturity across the pipeline (RQ2).

The main unresolved bottlenecks are limited interoperability between BIM schemas and rule ontologies, insufficiently verifiable reasoning in AI-assisted pipelines, and the absence of shared benchmark datasets that allow fair cross-study comparison. Trustworthy ACC therefore depends on transparent reasoning traces, open benchmark structures, and systematic human oversight rather than on further automation alone (RQ3).

AI and LLMs are beginning to connect regulatory text with BIM and IFC data through contextual interpretation, retrieval, semantic alignment, and hybrid reasoning support. However, the literature reviewed here indicates that their most reliable role at present is within grounded and verifiable pipelines rather than as standalone compliance engines. Their practical value lies in reducing manual effort in upstream tasks such as clause structuring, candidate rule generation, and semantic support, while final assurance still depends on symbolic validation, provenance capture, and expert review.

Methodologically, the AI-assisted PRISMA workflow also shows that evidence synthesis in ACC can move beyond a static review model toward a more updateable and traceable process. Substantively, the field now needs shared ontology-linked corpora, interoperable benchmark structures, hybrid symbolic–neural reasoning engines with explicit provenance, and evaluation protocols that combine technical performance with explainability and human factors. Advancing these priorities is essential if AI-driven ACC is to move from promising prototypes and controlled demonstrations to reliable, interpretable, and broadly deployable tools for AEC practice.

CRediT authorship contribution statement

Youssef Senousy: Writing – original draft, Visualization, Validation, Software, Methodology. **Franco Cheung:** Writing – review & editing, Supervision. **Thomas Beach:** Writing – review & editing. **Ogerta Elezaj:** Supervision. **Edlira Vakaj:** Supervision, Project administration.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Appendix A. ChatGPT deep search protocol and provenance checks

Run date and archived outputs. All ChatGPT Deep Search query cycles were executed on 28 July 2025. For each cycle, the returned candidate list was exported in CSV format and archived in the project repository for screening and traceability.

Base prompt template. Across all cycles, a fixed prompt structure was used, changing only the lens term(s) paired with the anchor phrase ACC and the publication window (2022–2025). The prompt requested a CSV output with the following fields: title, authors, year, DOI or URL, abstract, and source (journal or conference).

Template prompt.

> Search the web for academic papers on: "Automated Compliance Checking" AND [LENS] published between 2022 and 2025. Return a structured list with: Title, Authors, Year, DOI or URL, Abstract, Source (journal/conference). Output as CSV.

Deep search clarification responses. Deep Search requested clarification about scope and breadth. In Cycle 1, an inclusive option was selected for maximum lead discovery (All sources and a broad list). In Cycles 2 to 8, retrieval was restricted to peer reviewed journal and conference papers and a comprehensive list was requested.

Query cycles and archived CSV exports. Table A.7 reports the eight Deep Search cycles, their lens terms, the scope response, and the archived CSV output used as input to screening.

Note on output heterogeneity. Deep Search outputs may include candidate leads that fall outside conventional indexing channels, including preprints (for example SSRN or arXiv) and non publisher access points hosted on repositories or aggregator sites. For this reason, all Deep Search outputs were treated strictly as candidate leads and were subject to independent provenance verification before any inclusion decision.

Provenance and eligibility gate for deep search candidates. Each Deep Search candidate record was processed through the same de duplication and screening workflow as database derived records. A candidate was retained only if all of the following conditions were met:

1. Verifiable identity: bibliographic identity could be corroborated through a stable identifier, typically a DOI, and or trusted publisher or indexing entries.
2. Peer review eligibility: the venue satisfied the study eligibility criteria for peer reviewed journal or conference publications.
3. Full text access: the full text was accessible for assessment against the predefined inclusion and exclusion criteria.

Records lacking verifiable identifiers or reliable source traces, or remaining preprint only without a confirmable peer reviewed counterpart, were excluded during screening.

Integration into the PRISMA screening workflow. After export and archiving, Deep Search candidates were merged with database exports, de duplicated, and screened using the same title and abstract screening and full text assessment procedures described in Section 3.4. Inclusion decisions and full text exclusions were logged in the structured screening sheet stored in the project repository.

Appendix B. AI-assisted PRISMA algorithms

Algorithm 1 accepts a set of PDF files $P = p_1, p_2, \ldots, p_n$ and produces a corresponding set of structured text files $T = t_1, t_2, \ldots, t_n$. For each document p_i, the process begins by extracting the primary textual content using PDFMiner. If a file lacks an embedded text layer (e.g., a scanned document), OCR is applied after converting the pages to images. Tables are then extracted with pdfplumber, and

Table A.7
ChatGPT Deep Search query log and archived CSV exports (run date: 28 July 2025).

Cycle	Lens term(s) paired with ACC	Scope response	Archived CSV export
1	AEC	All sources; broad list	01_ACC_(2022–2025)_AND_AEC_11.csv
2	Natural Language Processing	Peer reviewed; comprehensive	02_ACC_(2022–2025)_AND_NLP_10.csv
3	BIM OR IFC	Peer reviewed; comprehensive	03_ACC_(2022–2025)_AND_BIM_OR_IFC_10.csv
4	Rule Based	Peer reviewed; comprehensive	04_ACC_(2022–2025)_AND_Rule_Based_12.csv
5	Ontology	Peer reviewed; comprehensive	05_ACC_(2022–2025)_AND_Ontology_12.csv
6	Knowledge Graph	Peer reviewed; comprehensive	06_ACC_(2022–2025)_AND_Knowledge_Graph_6.csv
7	Large Language Models	Peer reviewed; comprehensive	07_ACC_(2022–2025)_AND_Large_Language_Models_10.csv
8	Artificial Intelligence OR AI	Peer reviewed; comprehensive	08_ACC_(2022–2025)_AND_Artificial_Intelligence_OR_AI_10.csv

Algorithm 1: PDF-to-text

Input: $P = \{p_1, p_2, \dots, p_n\}$: Set of PDFs
Output: $T = \{t_1, t_2, \dots, t_n\}$: Set of structured text files

foreach $p_i \in P$ **do**
  Text Extraction:;
  $M_i \leftarrow$ PDFMiner(p_i);
  if $M_i = \emptyset$ **then**
    $M_i \leftarrow$ OCR(convert_to_images(p_i));
  Table Extraction:;
  $B_i \leftarrow$ extract_tables(p_i);
  Image Extraction:;
  $I_i \leftarrow$ extract_images(p_i);
  OCR on Images:;
  $O_i \leftarrow \sum_{\text{img} \in I_i}$ OCR(preprocess(img));
  Cleaning:;
  clean(M_i, B_i, O_i);
  Combine into Structured Text:;
  t_i = [TEXT: M_i; TABLES: B_i; IMAGES: I_i; OCR TEXT: O_i];
return $T = \{t_1, \dots, t_n\}$;

Algorithm 2: Semantic ranking to themes

Input: $D = \{d_1, d_2, \dots, d_n\}$: Structured text files from Algorithm 1; $T = \{\tau_1, \tau_2, \dots, \tau_m\}$: Predefined theme descriptions; ω: Weight vector for sections (*text*=1.0, *tables*=0.3, *images*=0.2); Thresholds: $\theta_{direct} = 0.70$, $\theta_{partial} = 0.50$.
Output: Matrix $R \in \mathbb{R}^{n \times m}$: similarity scores; annotated CSV with relevance labels.

Embed Themes:;
foreach $\tau_j \in T$ **do**
  $E_{\tau_j} \leftarrow$ SentenceTransformer(τ_j);

foreach $d_i \in D$ **do**
  Partition d_i **into sections:** $\{s_{text}, s_{tables}, s_{images}\}$;
  Split sections into sentences where $|u| \geq 8$ words;
  Initialise $max_score[\tau_j] = 0$ and $top_sentence[\tau_j] = \emptyset$ for all τ_j;
  foreach *section* s_k *with weight* ω_k **do**
    foreach *sentence* $u \in s_k$ **do**
      $e_u \leftarrow$ SentenceTransformer(u);
      foreach $\tau_j \in T$ **do**
        $sim \leftarrow \text{cosine}(e_u, E_{\tau_j}) \times \omega_k$;
        if $sim > max_score[\tau_j]$ **then**
          $max_score[\tau_j] \leftarrow sim$;
          $top_sentence[\tau_j] \leftarrow u$;
  Assign Relevance Label for each τ_j:;
  if $max_score[\tau_j] \geq \theta_{direct}$ **then**
    $label \leftarrow$ "direct";
  else if $\theta_{partial} \leq max_score[\tau_j] < \theta_{direct}$ **then**
    $label \leftarrow$ "partial";
  else
    $label \leftarrow$ "not addressed";
  Store results for d_i in R with labels and *top_sentence*;
Export R;

embedded images are retrieved using PyMuPDF. OCR is performed on these images after basic preprocessing (greyscale conversion and thresholding) to extract text from figures. The extracted components—main text M_i, tables B_i, image references I_i, and OCR text O_i—are cleaned to remove noise and merged into a structured output file containing four labelled sections: TEXT, TABLES, IMAGES, and OCR TEXT. This final text representation forms the basis for subsequent semantic analysis and ranking. To evaluate each paper against the main directions of current ACC research, the themes are used as target vectors in the semantic-ranking procedure described in Algorithm 2.

Algorithm 2 takes as input the corpus of structured text files $T = \{t_1, t_2, \dots, t_n\}$ produced in Algorithm 1 and returns, for every article t_i, a vector of thematic labels $L_i = \langle \ell_i^1, \ell_i^2, \dots, \ell_i^{10} \rangle$ together with the associated similarity scores $S_i = \langle \sigma_i^1, \sigma_i^2, \dots, \sigma_i^{10} \rangle$.

To initialise the procedure, each research theme mentioned above θ_j ($j = 1, \dots, 10$) is represented by a reference embedding $e(\theta_j)$ derived from its narrative description and keyword list. The 384-dimensional vectors form the anchor set $E = e(\theta_1), e(\theta_2), \dots, e(\theta_{10})$.

For each document $t_i \in T$, the algorithm first divides the content into three labelled sections: main body M_i, tables B_i, and OCR-derived figure text O_i. Each section is then tokenised into sentences, with sentences shorter than a predefined length discarded. Every remaining sentence s is embedded to obtain $e(s)$. Cosine similarity is calculated between $e(s)$ and each theme vector $e(\theta_j)$. The resulting score is multiplied by a section-specific weight—$w(M) = 1.0$ for the main body, $w(B) = 0.3$ for the extracted tables, and $w(O) = 0.2$ for the OCR-derived image content—to reflect the relative semantic richness of each source. For every theme θ_j, the algorithm records the maximum weighted similarity and stores the sentence s that achieves this value.

$$\sigma_{ij} = \max_{s \in \{M_i, B_i, O_i\}} \left[w(\text{section}(s)) \cdot \cos(e(s), e(\theta_j)) \right],$$

Each document t_i is assigned a theme-specific label ℓ_{ij} based on its maximum similarity score σ_{ij} with respect to each theme θ_j. If the score is 0.70 or higher, the article is considered to "directly addressed" the theme. If the score falls between 0.50 and 0.70, the article is considered to "partially addressed" the theme. If the score is below 0.50, the article is considered to "not addressed" the theme. This process results in the thematic profile (L_i, S_i) for article t_i.

$$\ell_{ij} = \begin{cases} \text{"directly addressed"}, & \sigma_{ij} \geq 0.70, \\ \text{"partially addressed"}, & 0.50 \leq \sigma_{ij} < 0.70, \\ \text{"not addressed"}, & \sigma_{ij} < 0.50. \end{cases}$$

Table B.8
Parameter grid used for sensitivity analysis.

Setting	θ_{direct}	$\theta_{partial}$	ω_{text}	ω_{tables}	ω_{images}
Baseline	0.70	0.50	1.00	0.30	0.20
Stricter thresholds	0.75	0.55	1.00	0.30	0.20
Looser thresholds	0.65	0.45	1.00	0.30	0.20
Conservative multimodal weights	0.70	0.50	1.00	0.20	0.10
Stronger multimodal weights	0.70	0.50	1.00	0.40	0.30

Table B.9
Sensitivity outcomes at paper level.

Setting	Direct papers	Partial-only	Overlap (%)	Jaccard	Theme stability
Baseline	78	10	100.0	1.000	Reference
Stricter	60	28	76.9	0.769	Top themes unchanged; minor rank swap
Looser	85	3	100.0	0.918	Top themes unchanged; minor rank swap
Conservative	78	10	100.0	1.000	Top themes unchanged
Stronger	78	10	100.0	1.000	Top themes unchanged

Table B.10
Sensitivity outcomes at theme-assignment level (totals across 88 papers ×10 themes = 880 labels per setting).

Setting	Direct labels	Partial labels	Not addressed labels	Top-5 themes
Baseline	248	605	27	TD; FRT; RRR; SA; EB
Stricter	99	705	76	TD; FRT; SA; RRR; EB
Looser	470	401	9	TD; FRT; EB; RRR; SA
Conservative	248	605	27	TD; FRT; RRR; SA; EB
Stronger	248	605	27	TD; FRT; RRR; SA; EB

Abbreviations: TD = Tool Development and Real-World Application; FRT = Formalisation of Regulatory Text; RRR = Rule Representation and Reasoning; SA = Semantic Alignment with BIM/IFC; EB = Evaluation and Benchmarking.

B.1. Parameter calibration, and sensitivity

Weights. The section-weight vector ω is applied to limit over-reliance on extraction channels that are less consistently available across papers and more sensitive to parsing noise. Narrative text is present in almost all papers and provides the most reliable semantic signal; therefore $\omega_{\text{text}} = 1.0$. Extracted tables and OCR-derived figure text can provide additional evidence but are more susceptible to extraction errors; they are consequently down-weighted (baseline: $\omega_{\text{tables}} = 0.3$, $\omega_{\text{images}} = 0.2$). The weights only guide how papers are prioritised and grouped during screening; final inclusion is still decided from full-text review. In our corpus, changing the non-text weights did not change which papers or themes were prioritised, because the strongest semantic matches usually came from the narrative text (Tables B.9 and B.10).

Thresholds. Thresholds were set to create two levels of relevance. Papers with a score above θ_{direct} are treated as *direct* matches (high confidence). Papers with a score between $\theta_{partial}$ and θ_{direct} are treated as *partial* matches and kept for manual checking rather than being excluded too early. Sensitivity results confirm the expected behaviour: stricter thresholds shift borderline cases from *direct* to *partial*, increasing manual screening load, whereas looser thresholds promote additional borderline papers into the *direct* pool without removing baseline direct papers (Table B.9).

Sensitivity analysis. To test whether screening outcomes depend on a single manual parameter configuration, we varied thresholds and section weights over plausible ranges and re-ran Algorithm 2. The tested parameter grid is shown in Table B.8.

For each configuration, we recorded paper-level pool membership (*Direct* vs. *Partial-only*) together with overlap against the baseline direct set (overlap and Jaccard), and theme-assignment outcomes across the full paper–theme matrix (total *directly addressed*, *partially addressed*, and *not addressed* labels), alongside the stability of the dominant themes. Paper-level results are summarised in Table B.9, while theme-level label totals and the top themes are reported in Table B.10. Here, overlap and Jaccard are computed on the set of *Direct* papers: Overlap $= \frac{|S \cap B|}{|B|} \times 100$ and $J = \frac{|S \cap B|}{|S \cup B|}$, where B denotes the baseline direct set and S denotes the direct set under the tested configuration; for example, under the looser setting, $|B| = 78$, $|S| = 85$, and $|S \cap B| = 78$, giving Overlap $= 100\%$ and $J = \frac{78}{85} = 0.918$.

Overall, threshold variation produced predictable strictness effects (e.g., the stricter setting reduced the number of direct papers and increased the partial-only pool), while the dominant thematic structure remained stable across all runs (top themes unchanged, with at most minor rank swaps). Notably, weight variation (conservative vs. stronger multimodal settings) produced identical paper-level pool counts and the same top themes as the baseline (Tables B.9 and B.10), suggesting that in this corpus the strongest semantic evidence is typically captured in narrative text and that the synthesis is robust to reasonable multimodal weighting choices (see Table B.11).

Data availability

The review data and supporting materials generated during this study, including the structured screening sheet, complete paper–theme similarity and ranking matrix, ChatGPT Deep Search query logs and archived CSV exports, and supporting project files, are available in the project GitHub repository: https://github.com/ysenousy/ACC-N-ACE.

Table B.11

Thematic papers matrix — Multimodal information extraction						
Authors	Methodology	Tools	Domain	Contributions	Theme score	Theme relevance (R/PR/NR)
J. Wu et al. (2025) [109]	Graph-based propagation, sensitivity analysis, and conditioned sampling generate compliant design variants. The framework adapts BIM models post-checking while preserving topology and minimising deviation from the original design.	Autodesk Revit, Dynamo API, NetworkX, Morris sensitivity analysis, conditioned Latin hypercube sampling, and a custom Revit-based code checker.	BIM for automated architectural code compliance under International Building Code spatial rules.	Automates design correction using ACC results, applies graph reasoning for local fixes, combines SA with sampling, and supports designer-centred, constraint-aware model adaptation.	0.7340	NR
Yu. Zhou et al. (2022) [69]	Integrates natural language processing and context-free grammar to interpret regulatory text. Uses deep learning for semantic labelling and CFG-based parsing to build computable syntax trees.	Python, PyTorch, BERT, ANTLR4 parser, Revit Model Checker, XML rule generation, and GitHub open dataset for validation.	Automated rule and code compliance checking in BIM for regulatory interpretation and rule creation.	Introduces NLP-CFG hybrid framework for automated rule interpretation, creates the first open regulation dataset, achieves 99.6% and 91% parsing accuracy, and supports explainable, multi-requirement rule extraction.	0.6930	R
T. Beach et al. (2024-2) [73]	Combines literature review, regulatory analysis, and proof-of-concept testing. Evaluates UK building regulations and develops two operational-phase demonstrators using lidar and computer vision.	RPLidar S1, Raspberry Pi, YOLOv5, Mask-R-CNN, TensorFlow, Python GUI, Revit integration, and UK Approved Documents database.	Operational-phase automated compliance checking for non-domestic buildings under UK regulations, focusing on fire safety and building monitoring.	Extends compliance automation to building operation, creates lidar and AI-based fire safety prototypes, validates feasibility through industry review, and defines use cases for whole life-cycle checking.	0.6750	R
S. Fuchs et al. (2022) [71]	Trains a transformer-based semantic parser to translate building code text into LegalRuleML. Uses data preprocessing, data augmentation, and multi-task learning for model refinement.	T5, BART, RoBERTa, Hugging Face Transformers, PDFMiner, Python, and LegalRuleML corpus from the New Zealand Building Code.	Automated regulatory translation and compliance checking in BIM using semantic parsing of building codes.	Introduces end-to-end neural translation of regulations, creates an annotated NZ Building Code corpus, achieves 41% F1 accuracy, and demonstrates machine-assisted rule encoding for ACC.	0.6660	R
R. Zhang et al. (2022) [70]	Develops a deep learning method using bidirectional LSTM + MLP with semi-supervised learning to extract semantic relations and build hierarchical representations of building-code requirements.	TensorFlow, Python, GloVe embeddings, GPU (Tesla K80), and corpora from IBC, IECC, and ADA standards for labelled and unlabelled training data.	Automated compliance checking and regulatory text interpretation in the AEC domain through semantic relation extraction and requirement hierarchy modelling.	Introduces the "requirement hierarchy" representation, defines new semantic relation types, achieves 94% accuracy, and enables scalable, domain-adaptive rule translation using semi-supervised learning.	0.6610	R
S. Fuchs et al. (2024) [72]	Trains transformer-based T5 models for semantic parsing of building codes into LegalRuleML using reversible and lossy intermediate representations, paraphrasing, and self-reflection strategies to enhance accuracy and training efficiency.	T5 transformer, TensorFlow, Python, Hugging Face, LegalRuleML dataset, LRML editor, and GitHub repository for training scripts and evaluation data.	Automated compliance checking and semantic parsing of building regulations in the Architecture, Engineering, and Construction (AEC) domain.	Introduces intermediate representations to improve parsing by up to 6.6% F1, reduces training time by 75%, and enables interpretable, human-in-the-loop translation of regulations into formal LRML rules.	0.6600	R
Thematic papers matrix — Semantic alignment with BIM/IFC						
Authors	Method	Tools	Domain	Contributions	Theme score	Theme relevance (R/PR/NR)
L. Jia et al. (2025) [74]	Combines semantic web and IFC data through ontology construction, rule mapping, and SPARQL querying. Uses Jena reasoning to align building ontologies and automate compliance validation.	Apache Jena, Protégé, IFC-to-RDF converter, Fuseki server, SPARQL query interface, and Chinese BIM standards GB/T 51269-2017 and JG/T 151-2015.	Automated BIM-based code compliance checking for architectural design inspection using semantic web technologies and IFCOWL data.	Creates a reusable building ontology, designs Jena-based mapping and query rules, integrates SPARQL rule templates, and demonstrates ontology-driven automated code checking with IFC models.	0.8150	R
Z. Ma et al. (2024) [75]	Uses ontology-based modelling with OWL and SWRL to represent BIM quality standards and execute automatic compliance checking on IFC models through rule-based inference.	Protégé, Apache Jena, OWL API, BIMServer for visualisation, IFC-to-ontology mapping scripts, and a BIM-QC ontology and ruleset in SWRL.	Automated quality compliance checking of BIM models under Chinese national BIM quality standards using ontology technology.	Proposes BIM-QC ontology and rulebase for real BIM quality standards, maps IFC entities to ontology, develops a prototype for automated checking, and validates scalability with a large IFC model.	0.7860	R

(*continued on next page*)

Table B.11 (*continued*).

D. Chen et al. (2024) [110]	Develops a three-step AI pipeline for fire risk management: computer vision interprets building blueprints, a graph algorithm assesses fire risks, and a diffusion model redesigns unsafe layouts.	Python, OpenCV, PyTorch, NetworkX, Tesseract OCR, CLAHE, Dijkstra algorithm, and the RPlan dataset; implemented on GPU with Adam optimiser.	Fire risk assessment and mitigation in architectural design using raw 2D blueprints instead of BIM models.	Introduces blueprint-based fire risk assessment and mitigation, automates layout redesign via diffusion models, eliminates BIM dependency, and enables rapid, full-cycle fire safety checking.	0.7690	NR
S. Zhu et al. (2024) [77]	Builds a BIM model quality compliance system using knowledge graph technology. Extracts BIM standard knowledge, structures it via NLP, and performs automated checking through a Revit C# plug-in and SQL rule base.	Revit 2020, Visual Studio 2019, C#, SQL Server 2019, Neo4j graph database, Python NLP modules, and Chinese BIM standards (GB/T 51212-2016 to GB/T 51447-2021).	BIM model quality compliance checking for Chinese building projects, focusing on modelling accuracy, data integrity, and delivery standards.	Creates a soft-coded knowledge graph rule base, automates BIM quality checking within Revit, verifies accuracy through a metro-station case, and achieves full correction of model errors.	0.7680	R
P. Hagedorn et al. (2025) [76]	Uses the Linked Open Terms (LOT) approach to design a multi-layered ontology framework integrating process, regulatory, and document ontologies. Employs SHACL, SPARQL, and ICDD for validation and data exchange.	Apache Jena, Protégé, PyLODE, Pellet Reasoner, OOPS! Scanner, IFCOWL converter, ICDD platform, SHACL engine, SPARQL, and GitHub repository for ontology publication.	Digital building permit review and automation across regulatory, procedural, and technical domains using semantic web and ontology integration.	Introduces OntoBPR, the first ontology-based framework unifying building permit processes. Aligns multiple domain ontologies via ICDD and SHACL, enables automated review, and standardises digital permitting workflows.	0.7660	R
E. Makisha (2023) [78]	Develops a RuleML-based algorithm to translate regulatory documents into machine-readable rules, using logical tagging, IFC mapping, and Excel-based automation for rule generation and classification.	RuleML, IFC 4.2, Excel VBA environment, Visual Basic for Applications, IFC terminology mapping, and Russian Code SP 54.13330.2016 dataset.	Automated BIM model verification and regulatory translation for residential building standards in the Russian Federation.	Creates a formalised RuleML rule-generation algorithm, links text logic to IFC data, classifies 356 code statements, and offers recommendations to improve regulatory structure for automation.	0.7650	R
Thematic papers matrix — Formalisation of regulatory text						
R. Zhang and N. El-Gohary (2022) [63]	Develops an intelligent building code framework using deep learning and natural language generation. Uses RNN-based encoder–decoder models with attention and copying mechanisms to generate natural-language requirements from semantic hierarchies.	TensorFlow, Python, NLTK, Google Colab (Tesla K80 GPU), IBC/IECC/ADA corpora, BLEU and ROUGE metrics, and annotated datasets of requirement hierarchies.	Automated regulatory reasoning and intelligent code generation for building regulations within the Architecture, Engineering, and Construction (AEC) domain.	Introduces the multi-form semantic (MFS) requirement hierarchy, proposes an RNN-based method for intelligent code generation, achieves 94% BLEU accuracy, and improves semantic linking precision to 92%.	0.7660	R
S. Zentgraf et al. (2022) [64]	Uses the Linked Open Terms (LOT) and Ontology Requirements Specification Document (ORSD) frameworks to design and validate an ontology for building code data. Demonstrates SHACL-based property validation.	Protégé, Apache Jena, RDF/OWL, SHACL, SPARQL, QUDT unit catalogue, and ISO 23386 property schema; ontology released as the Interconnected Data Dictionary Ontology (IDDO).	Ontology-based representation of building code information for BIM and Linked Building Data (LBD) workflows.	Introduces the IDDO ontology aligned with ISO 23386, links BOT, OPM, and BPO standards, supports SHACL rule validation, and enables FAIR-compliant automated property management.	0.7600	NR
Z. Zheng et al. (2022) [65]	Proposes a knowledge-informed NLP framework for automated rule checking. Combines ontology-based modelling, unsupervised semantic alignment, conflict resolution, and SPARQL code generation for rule interpretation.	Protégé, Python, GraphDB, Gensim (Word2Vec), BERT, ANTLR4, IFCOpenShell, SWRL.	Automated rule interpretation and compliance checking in BIM-based fire safety design.	Develops FPBO ontology, introduces unsupervised alignment with 90.1% accuracy, automates SPARQL rule generation, accelerates rule interpretation fivefold.	0.7590	R
D. Lehoczky et al. (2022) [66]	Adapts the 3S cognitive psychology model to study human trust in automated rule checking (ARC). Conducts an online experiment with Danish AEC professionals using manipulated BIM rule-checking interface screenshots.	Solibri Model Checker (BART plug-in), Danish Building Regulations BR18 dataset, online survey platform, Fisher's exact test, and statistical analysis via contingency tables and t-tests.	Socio-technical evaluation of human–automation trust in BIM-based automated rule checking within the Danish construction industry.	identifies information features influencing user confidence, distinguishes semantic vs. source-based trust behaviours, and provides design recommendations for more trustworthy ARC systems.	0.7560	NR

(*continued on next page*)

Table B.11 (*continued*).

J. Lee et al. (2023) [67]	Defines high-level implementable methods to translate natural-language building codes into computable logic. Decomposes clauses into atomic sentences, extracts verb phrases, and encodes them as reusable pseudo-code methods.	Python, KBIMCode, KBIMLogic, KBIMAssess, JSON/XML rule formats, context-free grammar parser, and a 200-dimensional word-embedding model for method clustering.	Automated building permit checking in BIM under the Korean Building Act using natural language processing and logic-based rule encoding.	Introduces 60 high-level rule methods, creates a hierarchical method taxonomy, builds the KBIMCode and KBIMLogic tools for code translation, and demonstrates full BIM-linked automated compliance checking.	0.7560	PR
J. Wu (2023) [68]	Develops the I-SNACC framework combining invariant signatures, logic reasoning, and NLP. Automates rule extraction, model validation, logic execution, and compliance reporting through modular integration and iterative testing.	Java, Python, B-Prolog, IFCOpenShell, APSTEX IFC toolkit, NLP libraries, GUI modules, invariant signature algorithms.	Automated BIM-based building code compliance checking for spatial and quantitative requirements under the International Building Code (IBC 2015).	Creates I-SNACC, the first fully modular ACC system; integrates rule generation, validation, and reasoning; achieves 95.2% precision and 100% recall; and cuts manual refinement time by 45.4%.	0.7560	R
Thematic papers matrix — Rule representation and reasoning						
T. Beach et al. (2024) [34]	Designs and validates an ecosystem architecture of distributed compliance-checking services using open APIs. Implements and tests the system on UK regulations through a prototype integrating multiple data and rule engines.	Drools Rule Engine, JSON/XML APIs, Solibri Office, SBEM, Netflix Eureka, BRE Templater, RASE markup, and GitHub-hosted ecosystem components.	Automated compliance checking and digital permitting for UK non-domestic buildings covering energy, accessibility, and fire safety.	Proposes scalable compliance-checking ecosystem, achieves 85% automation coverage, validates accuracy against manual checks, and demonstrates extensibility beyond BIM-based monolithic systems.	0.7650	NR
J. Wu and J. Zhang (2022) [84]	Introduces a BIM model validation method using invariant signatures and logic inference. Classifies building code concepts into four types and develops heuristic algorithms for concept detection and mapping.	IFC2x3 models, Python, B-Prolog, heuristic rule engine, logic fact generator, and datasets from IBC 2015 Chapter 10 across six real projects.	Automated BIM model validation and building code compliance checking under the International Building Code (IBC 2015).	Creates invariant signature algorithms for model validation, achieves 99.8% precision and 99.6% recall, defines four concept types, and bridges BIM data with regulatory concepts through logic-based inference.	0.7600	R
M. Yang et al. (2023) [86]	Develops a knowledge-graph-based model (DCKG) for semi-automatic design code representation. Defines four schemas—order, complex, event, and integration—to represent multi-level semantics and builds an annotation platform for clause interpretation.	Neo4j, Python, ECharts, Spring Boot, IFC models, GB50157-2013 Metro Design Code, and GitHub dataset (yang-mingsong/dckg4acc).	ACC in BIM for complex architectural and infrastructure design codes.	Proposes the DCKG model for high-complexity code representation, introduces four semantic schemas, builds an expert-guided annotation platform, and outperforms prior models in semantic completeness and accuracy.	0.7600	R
F. Yang and J. Zhang (2024) [87]	Develops a prompt-based framework using GPT-4 for automatic translation of building code text into logic programming language. Evaluates precision, recall, and efficiency on IBC 2015 and crash report datasets.	ChatGPT-4 API, Python 3.9, Picat logic programming language, regular-expression debugging scripts, and evaluation metrics (Jaro–Winkler, Jaccard, Levenshtein, precision, recall).	Automated building code compliance checking and information transformation in BIM and safety engineering contexts.	Combines GPT models with prompt engineering for code translation, achieves 97.37% precision and 95.88% recall, reduces generation time by 27.8%, and enables full automation with minimal training data.	0.7520	R
P. Patlakas et al. (2024) [88]	Integrates finite element analysis (FEA) with a semantic web framework for automated compliance checking. Builds a Building Compliance Ontology (BCO) and applies SHACL validation on IFC-enriched structural data.	Code_Aster, IFC2CA, IFC2BCO, Protégé, SHACL, Docker, Meshio, Python, and IFCOWL for ontology integration with Eurocode 3 structural rules.	Automated compliance checking in structural engineering, focusing on modular steel buildings under Eurocode standards.	Proposes the first ontology–FEA integrated ACC system, creates BCO for statement-based regulation modelling, enables IFC–FEA interoperability, and validates automation on a real modular housing case.	0.7520	R

(*continued on next page*)

Table B.11 (*continued*).

X. Xue and J. Zhang (2022) [85]	Expands a pattern-matching regulatory information ruleset for building codes using NLP and ontology matching. It iteratively generates and validates new rules to transform natural-language clauses into computable logic.	Python, GATE (ANNIE POS tagger), B-Prolog, IFC models, ontology editor, Levenshtein, Jaccard, Jaro–Winkler metrics, and IBC 2015 dataset (Chapters 5 and 10).	Automated compliance checking and rule transformation for building regulations in the Architecture, Engineering, and Construction (AEC) domain.	Develops a low-cost iterative ruleset expansion method, improves precision to 95.17% and recall to 96.60%, proves scalability across code chapters, and enables broader automated reasoning coverage.	0.7520	R
Thematic papers matrix — Integration of ontologies and knowledge graphs						
Z. Kabzhan et al. (2025) [83]	Combines ontological modelling and NLP to analyse regulatory documents. Constructs semantic profiles with subject–predicate–object–modality components to detect duplication, contradictions, and inconsistencies in construction standards.	Datavera EKG platform, Apache Fuseki, PostgreSQL, SKOS ontology, BFO, IFC-based domain ontology, SpaCy NLP library, and Python-based semantic analysis modules.	Automated regulatory document analysis and semantic modelling for construction industry digitalisation in Kazakhstan.	Develops ontology–NLP hybrid for semantic regulation analysis, detects contradictions with 0.86 F1 score, processes 242k-word corpus under 10 s, and supports BIM-integrated compliance automation.	0.7570	R
P. Zech et al. (2024) [79]	Develops a semantic web-based model-checking framework for IFC BIM models using Apache Jena. Implements rule- and query-based reasoning to validate specifications via an Eclipse plugin.	Apache Jena, OWL/SWRL, SPARQL, IFCOWL, Eclipse IDE plugin, Java, IFC2x3 and IFC4 models, and an open GitHub repository for rule specifications.	Automated compliance checking and BIM model validation through semantic reasoning and vendor-neutral ontology integration.	Creates open-source "BIMReason" framework, supports custom rule definitions, integrates reasoning with IFCOWL, validates across models, and achieves multi-version interoperability in reasonable computation time.	0.7370	R
R. Kruiper et al. (2024) [80]	Develops the I-REC platform for digitalising regulatory compliance. Integrates NLP, ML, and semantic web tools for requirement extraction, knowledge graph creation, and semantic search over 420 UK regulatory documents.	Python, SPAR.txt parser, GraphDB, Gensim, Uniclass, RDF, SKOS, Apache Jena, and a semantic search engine benchmarked against British Standards Online.	Digital regulatory compliance and automated building permit processing within UK construction regulations and safety standards.	Introduces platform-based NLP strategy for regulation digitalisation, achieves superior retrieval accuracy, creates automatic KG of 8k+ terms, and develops scalable semantic enrichment for regulatory corpora.	0.7290	R
A. Karmakar and V. Delhi (2025) [81]	Proposes a hybrid BIM semantic enrichment framework combining ML-based space classification and rule-based reasoning for tenement identification. Trains classifiers on IFC4 data and applies logic rules for grouping residential units.	Python, Scikit-learn, GraphSAGE, XGBoost, Revit, Dynamo plug-in, IFC4 models, Spearman correlation, and Mumbai DCPR 2034 dataset from 18 residential projects.	Automated tenement compliance checking and semantic enrichment in BIM for residential buildings during preconstruction permitting in Mumbai, India.	Integrates ML and rule-based reasoning for semantic enrichment, achieves 0.85 F1 accuracy, identifies tenements flawlessly, reduces preprocessing time, and enables scalable, user-verifiable automated compliance workflows.	0.7250	R
L. Jiang et al. (2022-2) [3]	Develops a BIM-based automated compliance checking framework using multi-ontology fusion. Integrates code, design, merging, and checking ontologies linked by mapping and checking rules through Apache Jena reasoning.	Apache Jena, Protégé, IFCtoRDF converter, SPARQL, Jena ARQ, Revit 2020, GB 50096 residential code, and an ontology-based checking platform.	Automated building code compliance checking in BIM using semantic web technologies for Chinese residential projects.	Creates a five-step ontology development roadmap, introduces a designed model ontology for efficiency, applies ontology mapping for semantic enrichment, and validates the platform on real BIM models with improved performance.	0.7210	R
W. Goh et al. (2024) [82]	Proposes a BIM-based ACC framework for structural calculations. Classifies design codes, defines variables in the IFC schema via property sets, and translates code conditions into SQWRL rules executed with Drools.	Revit 2020, IFCOWL converter, Protégé, Drools engine, OWL/SWRL/SQWRL, Java, Eclipse IDE, and the Korea Highway Bridge Design Code.	Automated compliance checking for structural engineering, focusing on prestressed concrete girder bridges under Korean design codes.	Develops transparent OpenBIM-based ACC framework, integrates design code logic into IFCOWL, verifies PSC bridge model accuracy	0.7160	R
Thematic papers matrix — Model-driven compliance intelligence						
N. Chen et al. (2024) [28]	Combines deep learning, large language models, and ontology reasoning for BIM compliance. Uses CNN-LSTM-BERT fusion for text classification, GPT-4 for rule extraction, and SPARQL for ontology-based verification.	Python, PyTorch, GPT-4 API, BERT-base-Chinese, Protégé, SPARQL, Revit-IFC converter, TF–IDF, and the GB50096-2011 residential design code.	Automated BIM compliance checking for Chinese residential building design using hybrid AI-ontology integration.	Introduces LLM–DL–ontology hybrid framework, achieves 0.99 accuracy in rule classification, automates rule extraction via GPT-4, maps results to ontology, and enables scalable, interpretable compliance checking.	0.6940	R

(*continued on next page*)

Table B.11 (*continued*).

H. Li et al. (2024) [89]	Develops an intelligent compliance-checking framework merging knowledge graphs and large language models. Uses BERT-BiLSTM for text parsing, CRF for entity extraction, and GPT-based inference guided by structured knowledge.	Python, PyTorch, BERT, CRF, ChatGPT-3.5, Neo4j, Cypher, Bridge Construction Standards (JTG/T 3650-2020, JTG F80/1-2023), and prompt-engineering templates.	Automated compliance checking of construction schemes for bridge engineering using AI-driven knowledge fusion.	Creates a hybrid KG-LLM method for construction scheme checking, achieves 72% accuracy, proves robustness to parsing errors, and enhances reasoning transparency through prompt templates and two-stage validation.	0.6700	R
R. Zhang and N. El-Gohary (2023) [11]	Proposes a transformer-based relation classification model for aligning IFC and regulatory concepts. Uses contextual embeddings, IFC knowledge graphs, and transfer learning for context-aware semantic matching.	Python, PyTorch, ALBERT/BERT, IFCOWL, RDF, SPARQL, IFC4 datasets, IBC, IECC, ADA standards, and Google Colab GPU (Tesla K80).	Automated context-aware semantic alignment of IFC-based BIM models with building regulations for compliance checking.	Introduces the first transformer-based IFC regulation alignment method, achieves 84.3% precision and 83.8% F1 score, integrates IFC definitions and knowledge graphs, and improves generalisability for multi-code applications.	0.6550	R
X. Wang and N. El-Gohary (2023) [90]	Develops a deep learning framework for relation extraction from safety regulations. Uses attention-based CNN and BiLSTM models with word and position embeddings to classify 56 relation types and build query graphs.	Python, TensorFlow, PyTorch, Stanford CoreNLP, CBOW and GloVe embeddings, OSHA 29 CFR 1926 dataset, RTX 2070 GPU, and custom query-graph generator.	Automated field compliance checking and safety requirement extraction in construction safety regulations, focusing on fall protection rules.	Creates CNN and BiLSTM models achieving 87.5% precision and 85.4% F1, builds query-graph representation for reasoning, introduces 56 domain relations, and automates requirement extraction for compliance checking.	0.6550	R
Y. Chen and H. Jiang (2024) [91]	Combines ontology and NLP to automate fire regulation interpretation. Uses a joint BERT-based entity–relation extraction model, context-free grammar parsing, and deontic logic transformation for rule formalisation and BIM rule checking.	Python, PyTorch, BERT, NLTK, Neo4j, Protégé, RDF2RDF, Revit Model Checker, CFG parser, GB50016-2014 Fire Code dataset, and Revit BIM models.	Automated compliance checking and spatial rule interpretation for fire safety design within BIM workflows.	Introduces Fire-GeoBERT model for ontology automation, achieves 86.27% F1 for entity extraction, 96.26% accuracy in logical rule formalisation.	0.6360	R
J. Zhang (2023) [92]	Compares ChatGPT-generated code with the rule-based SNACC system. Test the conversion of eight IBC requirements into Python logic, analysing accuracy, processing time, and handling of domain-specific terms.	ChatGPT (GPT-3), SNACC system, Python, Java timer functions, IBC 2006/2009 datasets, and logic program generation scripts.	Automated building code compliance checking and rule formalisation for architectural and structural requirements.	Demonstrates ChatGPT's ability to convert regulatory text to Python code, processes rules faster than SNACC, but lacks explicit unit handling.	0.6250	R
Thematic papers matrix — Explainability and trust in AI systems						
E. Nuyts et al. (2024) [7]	Performs a comparative analysis of eight compliance-checking approaches (IFC-, schema-, and linked-data-based) using five rule categories from Flemish accessibility codes to evaluate functional capacity, openness, and validation support.	Solibri Model Checker, JSON Schema, XSD, OWL, SWRL, SPARQL, SHACL, IFCOWL, Apache Jena, Protégé, Revit 2023, RDF/OWL datasets, and OSF data repository.	Automated compliance checking of construction data, comparing BIM, schema, and linked-data methods for accessibility rule representation and validation.	Provides the first cross-method benchmark for ACC tools, establishes reproducible datasets, and identifies SHACL as most suitable for standardised validation.	0.6980	R
K. Johansen et al. (2024) [94]	Develops a knowledge graph-based framework integrating safety analysis, 4D BIM, and digital twins. Uses Prolog for ontology reasoning and query generation to connect construction planning and safety data.	Prolog, IFC4D BIM model, Neo4j (optional), Apache Jena, Protégé, Revit 2023, and the SafeCon ontology shared via GitHub.	Automated construction safety and risk assessment within digital-twin environments for integrated planning, scheduling, and hazard mitigation.	Creates ontology and query framework for safety–planning integration, identifies reusable query building blocks, enables sub-second reasoning on 3800-node graphs, and enhances cross-domain safety decision-making.	0.6980	R
O. Doukari et al. (2022) [98]	Develops a bottom-up, object-centred ACC framework using simulated annealing principles. Encodes requirements as XML if–then rules, processes BIM objects hierarchically, and validates through two case studies with industrial object libraries.	Autodesk Revit 2020, C#, XML rule schema, Syneg Revit plug-in, FCSI material library, Excel reporting, Bonnet Thirode BIM library, and NBS National BIM Library.	Automated BIM object quality and compliance checking for manufacturing and building regulations across foodservice and fire safety domains.	Introduces bottom-up object-centred ACC method, satisfies all desired checking criteria, achieves 125-minute time savings per object, supports modular rule sets, and proves extensibility through real industrial validation.	0.6930	R

(*continued on next page*)

Table B.11 (*continued*).

S. Madireddy et al. (2025) [35]	Develops an LLM-driven BIM compliance framework integrating GPT-4, Claude, Gemini, and Llama to convert building codes into Python scripts for semi-automated rule checking and compliance reporting in Revit.	Revit 2024, IronPython 3.4.1, GPT-4, Claude 3.5, Gemini, Llama 3.1, Grok, Python API, IRC/IMC 2021 codes, and Autodesk Revit PythonShell.	Automated BIM-based code compliance checking and rule interpretation across architectural, structural, and mechanical regulations.	Creates a multi-model LLM framework for Revit, defines optimised prompt structure, achieves 76.7% script success (Grok), reduces manual checking time, and enables scalable, adaptable regulatory automation.	0.6870	R
H. Ying and R. Sacks (2024) [102]	Proposes an autonomous LLM-agent framework using GPT-4o for generic compliance checking. Combines RAG-based rule querying, ReAct agent reasoning, and BIM manipulation tools for automatic checking and design revision.	GPT-4o, RevitCopilot plug-in (C#/Python), Revit API, JSON data extraction, RAG text-embedding model, Unstructured.io, IBC 2009 Chapter 19 dataset, and BIM revision functions.	Autonomous building compliance checking and BIM design revision across architectural and structural domains.	Introduces first autonomous LLM-driven ACC system, integrates querying, checking, and revision, enables reasoning-and-acting workflow, demonstrates self-correcting capability, and automates Revit compliance.	0.6830	R
X. Xue et al. (2022) [85]	Develops a semi-automated method to extract tabular requirements from building codes and convert them into logic rules using header detection, layout recognition, and invariant signature-based information matching.	SQLite, Python, B-Prolog, IFC2x3 model, IBC 2015 tables, custom table-processing algorithms, invariant signature library, and a convenience store BIM model for validation.	Automated BIM-based code compliance checking focusing on International Building Code (IBC 2015).	Achieves 100% correct table extraction after refinement, generates 1542 logic rules from tabular data.	0.6730	NR
Thematic papers matrix — Human-in-the-loop approaches						
H. Hettiarachchi et al. (2025) [12]	Creates the Code-ACCORD dataset by semi-automatically extracting, filtering, and manually annotating self-contained regulatory sentences with entities and relations for rule-generation tasks.	Python, PDFMiner, NLTK, LightTag, Zenodo repository, and GitHub pipeline for annotation and curation; regulations sourced from UK and Finnish building codes.	Natural language processing (NLP) for automated compliance checking and regulatory text formalisation in the AEC sector.	Publishes the first open multilingual building code dataset with 862 annotated sentences, 4297 entities, and 4329 relations, enabling machine learning for rule extraction and automated compliance checking.	0.7060	R
J. Anim et al. (2024) [103]	Implements SHACL-SPARQL rules for automated legal compliance on RDF data. Extends the MELON legal ontology method with temporal and aggregate reasoning using the Time-Indexed Value in Context (TVC) pattern.	Apache Jena, SHACL-SPARQL, Protégé, RDF/OWL, DLV2 reasoner, TVC ontology, GitHub repository, and Ghana Petroleum Regulations (L.I. 2204) dataset.	Automated legal compliance checking in Ghana's petroleum sector using semantic web and ontology-based formalisation.	Introduces SHACL-SPARQL for aggregate and temporal reasoning, identifies legal gaps through ontology modelling, formalises 48 rules, and proves scalable, explainable compliance reasoning beyond OWL limits.	0.6860	PR
D. Al-Turki et al. (2024) [29]	Automates RASE annotation of building regulations using large language models with few-shot, fine-tuning, and progressive active learning to generate YAML-based rule files for compliance checking.	OpenAI GPT-4o, Python, OpenAI API, YAML (BCRL), SimGNN, JSONL datasets, Graph Edit Distance, and regulatory datasets from UK building codes.	Automated building regulation interpretation and rule formalisation in BIM-based compliance checking within the AEC industry.	Introduces a human-in-the-loop LLM system for RASE tagging, achieves high structural and text similarity via active learning, and demonstrates scalable, accurate rule extraction for automated compliance checking.	0.6640	R
T. Bloch et al. (2023) [105]	Applies graph neural networks (GNNs) for automated code checking. Generates synthetic BIM graphs, labels nodes for compliance, trains GAT and GCN models, and tests classifier accuracy on real residential floor plans.	Python, StellarGraph, TensorFlow, Neo4j, IFC-based topology graphs, ANSI A117.1 accessibility code, and BIM floor plan datasets from Israeli firms.	Automated compliance checking using machine learning in BIM, focusing on accessibility regulations for residential buildings.	Introduces first GNN-based ACC workflow, achieves 86.8% F1 score, validates on real BIM data, and demonstrates potential for scalable, relational compliance reasoning without explicit rule sets.	0.6450	PR

(*continued on next page*)

Table B.11 (*continued*).

V. Schuk et al. (2022) [104]	Analyses infrastructure regulations and proposes specifications for developing ACC tools. Defines 16 modelling requirements and designs a graph-based ontology for regulation–rule–object interaction.	Graph database, ontology modelling, object-oriented programming, BPMN, DMN, IFC-Rail schema, and German DB AG guidelines for rail and road construction.	Automated code compliance checking in infrastructure planning and transport engineering (road and rail).	Identifies ACC requirements for infrastructure, introduces graph-database ontology for rule storage, classifies regulation complexity, and provides a structured roadmap for real-world ACC tool development in BIM environments.	0.6450	PR
H. Gao et al. (2022) [111]	Develops a computational geometric approach for BIM semantic enrichment to automate underground garage compliance checking. Extracts geometric data, infers missing attributes, and validates models through geometric reasoning and 3D visualisation.	Revit, IFCOpenShell, Unity, Java, Python, OBJ format converter, triangular mesh algorithms, and nine real underground garage BIM models.	Automated BIM compliance checking for underground garage design and spatial validation.	Creates a real-world tested prototype achieving 98.3% lane detection, enhances BIM semantics for geometry-driven rule checking, and identifies clear height noncompliance.	0.6300	NR
Thematic papers matrix — Evaluation and benchmarking						
Z. Wu et al. (2024) [108]	Integrates knowledge graphs with performance-based generative design (PGD). Encodes expert sustainability knowledge as reasoning rules to automate compliance and optimisation while reducing computation through threshold-controlled solution space narrowing.	Python, RDFLib, Protégé, EnergyPlus, Radiance, Rhino3dm, NSGA-II, SWRL, SPARQL, BOT and Brick ontologies, and Hong Kong Building (Planning) Regulations dataset.	Knowledge-informed generative design and automated compliance checking for sustainable building optimisation in early BIM-based design stages.	Introduces PGD-KG model linking design and regulation, embeds expert knowledge for reasoning-based evaluation, reduces computation time by 73.25%, and enables interpretable, regulation-aware optimisation of sustainable buildings.	0.7900	R
X. Liu and X. Zhu (2023) [107]	Implements GPT-3/3.5 models for automated compliance checking through prompt engineering. Conducts four structured tests comparing zero-, one-, and few-shot prompts and fine-tuned models on fire safety regulation clauses.	Python, OpenAI GPT-3/3.5 API, ChatGPT Playground, Health Technical Memorandum (HTM 05-02), confusion matrices, accuracy, precision, recall, and F1-score metrics.	Automated building design compliance checking using large language models for fire safety regulations.	Benchmarks GPT-based ACC performance, achieves up to 91% accuracy for fine-tuned models, quantifies results using confusion matrices, and identifies prompt structure as key to improving LLM-based compliance checking.	0.7570	R
L. Robaldo et al. (2023) [106]	Compares SHACL and DLV2 reasoners for RDF-based compliance checking. Implements legal norms as logical rules, encodes compensatory and defeasible norms, and evaluates both systems across synthetic RDF datasets of increasing size.	Apache Jena, TopBraid SHACL (v1.3.2), DLV2 reasoner, Fuseki SPARQL endpoint, ASP-Core-2 standard, RDF/Turtle datasets, and Linux environment (32 GB RAM, 12-core CPU).	LegalTech and semantic web compliance reasoning using RDF, SHACL, and Answer Set Programming (ASP) for normative systems.	First empirical RDF-based comparison of SHACL and DLV2, showing DLV2's superior speed and scalability, validating ASP's efficiency, and supporting RDF-compatible explainable compliance checking frameworks.	0.7430	R
Y. Li et al. (2023) [97]	Proposes a semantic code representation model based on constraint triples. Maps rule elements to IFCOWL concepts and automatically generates SWRL rules for BIM-based compliance reasoning.	Protégé, SWRL, SPARQL, IFCOWL, RDF, IFCtoRDF converter, and BIM Vision for visualisation using a metro station IFC model.	Automated compliance checking for metro design regulations using ontology-driven semantic representation and rule reasoning.	Introduces a five-part constraint-based model representing relationships and spatial logic, supports automatic rule generation, and distinguishes checking conditions from requirements.	0.7300	R
Q. Ren et al. (2024) [33]	Combines rule syntax parsing, deep learning-based entity extraction, and semantic distance models to check concrete dam construction quality through automatic scoring rather than binary judgments.	Python, BERT-BiLSTM-CRF, SimBERT, ANTLR parser, CRF, syntax tree builder, SL 677-2014 and SL 632-2012 standards, and Xiangjiaba Dam dataset.	Automated construction quality compliance checking for hydraulic engineering using NLP and semantic alignment.	Achieves 88.89% accuracy and 12.89% F1 improvement, introduces fine-grained compliance scoring, improves entity extraction via BERT-BiLSTM-CRF, and enables text-based semantic reasoning for dam construction quality assurance.	0.7260	R

(*continued on next page*)

Table B.11 (*continued*).

A. Karmakar et al. (2024) [112]	Applies large language models to extract entities and relationships from unstructured concession reports. Performs ontology extraction, entity disambiguation, and community detection to generate municipal knowledge graphs for compliance reasoning.	Python, Mistral-7B, Zephyr-7B, JSON output structure, Girvan–Newman clustering, PDF archival dataset from Mumbai Municipal Corporation.	Knowledge graph generation from unstructured text for municipal compliance and concession approval in urban planning.	Introduces LLM-based ontology extraction for concession data, identifies contextual proximities via graph clustering, and reveals decision patterns.	0.7180	NR
Thematic papers matrix — Tool development and real-world application						
E. Nuyts et al. (2023) [20]	Implements SHACL for automated validation of BIM models against legislative constraints. Defines value, relational, and mathematical rules from RASE-annotated regulations and integrates SHACL-Sparql for complex conditional statements.	Python, PySHACL, npm, React UI, IFCtoLBD converter, RDF/Turtle, SHACL, SPARQL, and Flemish Accessibility Code dataset.	Automated building code compliance checking using linked data and semantic web technologies.	Evaluates SHACL's suitability for ACC, creates transparent validation reports, defines composite constraint types, and demonstrates interpretability, open standards compliance, and model quality assurance.	0.8480	R
A. Ismail et al. (2023) [96]	Develops a user-friendly BIM-based automated code compliance checking system using semi-automated RASE rule translation, Revit-based Dynamo scripts, and expert validation for Malaysian fire safety codes.	Autodesk Revit, Dynamo visual programming, Python pseudocode, RASE semantic mark-up, OmniClass, Revit Classification Manager.	Automated compliance checking for Malaysian fire safety regulations within BIM design workflows.	Creates BIMSMACC prototype with Revit integration, enables expert-involved rule translation, supports native Revit data, achieves mean usability and accuracy scores of 4.77–4.96, and improves designer accessibility.	0.8380	R
I. Huitzil et al. (2022) [95]	Develops SmartNorms4BIM prototype integrating OWL2 ontologies with IFC models. Translates housing regulations into formal logic, enriches BIM data, applies reasoning with HermiT, and visualises compliance outcomes in text and 3D form.	Protégé, OWL2, HermiT DL Reasoner, IFC2x3, Uniclass, CGAL geometry library, Python, and 3D BIM visualizer.	Ontology-based automated compliance checking for residential habitability regulations within BIM environments.	Creates SmartNorms4BIM prototype, introduces dual text–visual reporting, links IFC data to ontological classes, handles geometric validation via CGAL.	0.8330	R
G. Recski et al. (2024) [100]	Creates the BRISE-Plandok corpus of German building codes. Combines expert–annotator collaboration, rule-based semantic parsing, and machine learning baselines to extract formal rules and evaluate attribute-level accuracy.	Python, Stanza, BERT, Logistic Regression, Decision Tree, DLV2, Potato, 4Lang semantic parser, JSON rule format, GitHub repository, and Vienna zoning documents dataset.	Automated rule extraction and formal representation of city-level building regulations for digital permitting in Vienna's e-government program.	Produces gold-standard rule corpus, compares ML and rule-based models with precision–recall–F1 metrics, reaches 93% precision, and supports interpretable, benchmarked compliance reasoning through open, reproducible datasets.	0.8280	PR
Y. Zhou et al. (2022) [99]	Develops a Design for Safety (DFS) semantic framework using NLP and BIM integration. Extracts safety rules from Chinese regulations via NLPIR, builds DFS ontology in Protégé, and maps BIM data through Dynamo–Cellfie for SWRL-based reasoning.	Revit, Dynamo, NLPIR, Protégé 5.5.0, Cellfie, OWL, SWRL, Drools inference engine, and Excel.	Design safety compliance checking for construction projects using ontology-based and NLP-driven BIM integration.	Introduces NLP–ontology–BIM framework, achieving 95% recall and 90% precision in compliance checks, improving interoperability, automation, and safety validation within design documents.	0.8230	R
R. Kruiper et al. (2024-2) [101]	Develops a linked-data knowledge graph to map regulatory terms to construction vocabularies. Uses NLP-based term extraction, TF–IDF domain classification, semantic similarity, and Uniclass–Wikidata linking to support human annotators.	Python, SPAR.txt shallow parser, BERT embeddings, GraphDB, RDF, SKOS, Neo4j, Louvain clustering, VocBench 3, and UK Merged Approved Documents corpus.	Knowledge graph construction and terminology alignment for automated compliance checking in UK building regulations.	Creates I-REC lexicon knowledge graph linking 2100 terms, supports annotation via visual graphs, speeds expert conceptualisation sixfold, and provides open resources for semantic rule development in compliance checking.	0.8160	PR

References

[1] Y. Zou, B.H.W. Guo, E. Papadonikolaki, J. Dimyadi, L. Hou, Lessons learned on adopting automated compliance checking in the AEC industry: A global study, J. Manag. Eng. 39 (4) (2023) 04023019, http://dx.doi.org/10.1061/JMENEA.MEENG-5051.

[2] C. Preidel, A. Borrmann, Towards code compliance checking on the basis of a visual programming language, Electron. J. Inf. Technol. Constr. 21 (2016) 402–421, https://www.itcon.org/2016/25.

[3] L. Jiang, J. Shi, C. Wang, Multi-ontology fusion and rule development to facilitate automated code compliance checking using BIM and rule-based reasoning, Adv. Eng. Inform. 51 (2022) 101449, http://dx.doi.org/10.1016/j.aei.2021.101449.

[4] M.A. Yamusa, J.A. Dauda, S.O. Ajayi, A.B. Saka, A.S. Oyegoke, W.A. Adebisi, Z.T. Jagun, Automated compliance checking in the AEC industry: A review of current state, opportunities and challenges, 2024, http://dx.doi.org/10.2139/ssrn.5053757, SSRN Preprint.

[5] C. Eastman, J.M. Lee, Y.S. Jeong, J.K. Lee, Automatic rule-based checking of building designs, Autom. Constr. 18 (8) (2009) 1011–1033, http://dx.doi.org/10.1016/j.autcon.2009.07.002.

[6] A.S. Ismail, K.N. Ali, N.A. Iahad, A review on BIM-based automated code compliance checking system, in: Proc. 2017 International Conference on Research and Innovation in Information Systems, ICRIIS, Langkawi, Malaysia, 2017, pp. 1–6, http://dx.doi.org/10.1109/ICRIIS.2017.8002486.

[7] E. Nuyts, M. Bonduel, R. Verstraeten, Comparative analysis of approaches for automated compliance checking of construction data, Adv. Eng. Inform. 60 (2024) 102443, http://dx.doi.org/10.1016/j.aei.2024.102443.

[8] R. Amor, J. Dimyadi, The promise of automated compliance checking, Dev. the Built Environ. 5 (2021) 100039, http://dx.doi.org/10.1016/j.dibe.2020.100039.

[9] P. Pauwels, S. Zhang, Semantic rule-checking for regulation compliance checking: An overview of strategies and approaches, in: Proceedings of the 32nd International Conference of CIB 1198W78, Eindhoven, the Netherlands, 27–29 October 2015, pp. 619–628, https://itc.scix.net/pdfs/w78-2015-paper-064.pdf.

[10] T.H. Beach, J.L. Hippolyte, Y. Rezgui, Towards the adoption of automated regulatory compliance checking in the built environment, Autom. Constr. 118 (2020) 103285, http://dx.doi.org/10.1016/j.autcon.2020.103285.

[11] R. Zhang, N. El-Gohary, Transformer-based approach for automated context-aware IFC-regulation semantic information alignment, Autom. Constr. 145 (2023) 104540, http://dx.doi.org/10.1016/j.autcon.2022.104540.

[12] H. Hettiarachchi, A. Dridi, M.M. Gaber, P. Parsafard, N. Bocaneala, K. Breitenfelder, G. Costa, M. Hedblom, M. Juganaru-Mathieu, T. Mecharnia, S. Park, H. Tan, A.-R.H. Tawil, E. Vakaj, CODE-ACCORD: A corpus of building regulatory data for rule generation towards automatic compliance checking, Sci. Data 12 (2025) 170, http://dx.doi.org/10.1038/s41597-024-04320-x.

[13] Z. Zhang, L. Ma, T. Broyd, Rule capture of automated compliance checking of building requirements: A review, Proc. Inst. Civ. Eng. – Smart Infrastruct. Constr. 176 (4) (2023) 224–238, http://dx.doi.org/10.1680/jsmic.23.00005.

[14] L. Shi, D. Roman, From standards and regulations to executable rules: A case study in the building accessibility domain, in: Proc 9th International Conference on Knowledge Engineering and Ontology Development (KEOD 2017) CEUR Workshop Proceedings, Vol. 1875, 2017, https://ceur-ws.org/Vol-1875/paper12.pdf.

[15] M. Dragoni, S. Villata, W. Rizzi, Combining NLP approaches for rule extraction from legal documents, in: Proc 1st Workshop on Technologies for Regulatory Compliance (TeReCom 2017), CEUR Workshop Proceedings, 2017, https://hal.science/hal-01572443.

[16] J. Zhang, N.M. El-Gohary, Semantic NLP-based information extraction from construction regulatory documents for automated compliance checking, J. Comput. Civ. Eng. 30 (2) (2016) 04015014, http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000346.

[17] J. Zhang, N. El-Gohary, Automated regulatory information extraction from building codes: Leveraging syntactic and semantic information, in: Proc. Construction Research Congress 2012: Construction Challenges in a Flat World, 2012, pp. 63–72, http://dx.doi.org/10.1061/9780784412329.063.

[18] N.M. El-Gohary, T.E. El-Diraby, Domain ontology for processes in infrastructure and construction, J. Constr. Eng. Manag. 136 (7) (2010) 730–744, http://dx.doi.org/10.1061/(ASCE)CO.1943-7862.0000178.

[19] J. Zhang, N.M. El-Gohary, Automated information transformation for automated regulatory compliance checking in construction, J. Comput. Civ. Eng. 29 (4) (2015) B4015001, http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000427.

[20] E. Nuyts, J. Werbrouck, R. Verstraeten, L. Deprez, Validation of building models against legislation using SHACL, in: Proc. 11th Linked Data in Architecture and Construction Workshop (LDAC 2023) Matera, Italy, CEUR Workshop Proceedings, 3633, 2023, pp. 164–175, http://hdl.handle.net/1854/LU-01HCHF9JK71QP7RQCEZCQK49WC.

[21] H. Wan, J. Zhang, Y. Chen, Exploring gen-AI applications in building research and industry: A review, Build. Simul. 18 (2025) 1251–1273, http://dx.doi.org/10.1007/s12273-025-1279-x.

[22] M. Alnuzha, T. Bloch, The role of machine learning in automated code checking: A systematic literature review, J. Inf. Technol. Constr. 30 (2025) 22–44, http://dx.doi.org/10.36680/j.itcon.2025.002.

[23] X. Wang, N. El-Gohary, Deep learning–based named entity recognition and resolution of referential ambiguities for enhanced information extraction from construction safety regulations, J. Comput. Civ. Eng. 37 (5) (2023) 04023035, http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0001064.

[24] R. Zhang, N.M. El-Gohary, A machine-learning approach for semantic matching of building codes and building information models (BIMs) for supporting automated code checking, in: H. Rodrigues, G. Morcous, M. Shehata (Eds.), Recent Research in Sustainable Structures, GeoMEast 2019. Sustainable Civil Infrastructures, Springer, Cham, 2020, http://dx.doi.org/10.1007/978-3-030-34216-6_5.

[25] T. Beach, Y. Rezgui, H. Li, T. Kasim, A rule-based semantic approach for automated regulatory compliance in construction, Expert Syst. Appl. 42 (12) (2015) 5219–5231, http://dx.doi.org/10.1016/j.eswa.2015.02.029.

[26] P. Pauwels, D. van Deursen, R. Verstraeten, W. Terkaj, T. Krijnen, J. Beetz, P. van den Helm, A semantic rule checking environment for building performance checking, Autom. Constr. 74 (2011) 49–63, http://dx.doi.org/10.1016/j.autcon.2010.11.017.

[27] W. Solihin, C. Eastman, Classification of rules for automated BIM rule checking development, Autom. Constr. 53 (2015) 69–82, http://dx.doi.org/10.1016/j.autcon.2015.03.003.

[28] N. Chen, X. Lin, H. Jiang, Y. An, Automated building information modeling compliance check through a large language model combined with deep learning and ontology, Buildings 14 (2024) 1983, http://dx.doi.org/10.3390/buildings14071983.

[29] D. Al-Turki, H. Hettiarachchi, M.M. Gaber, M.M. Abdelsamea, S. Basurra, S. Iranmanesh, H. Saadany, E. Vakaj, Human-in-the-loop learning with LLMs for efficient RASE tagging in building compliance regulations, IEEE Access 12 (2024) 185291–185306, http://dx.doi.org/10.1109/access.2024.3512434.

[30] C. Liang, T. Le, Y. Ham, B.R.K. Mantha, M.H. Cheng, J.J. Lin, Ethics of artificial intelligence and robotics in the architecture, engineering, and construction industry, Autom. Constr. 162 (2024) 105369, http://dx.doi.org/10.1016/j.autcon.2024.105369.

[31] P.E.D. Love, W. Fang, J. Matthews, S. Porter, H. Luo, L. Ding, Explainable artificial intelligence (XAI): Precepts, models, and opportunities for research in construction, Adv. Eng. Inform. 57 (2023) 102024, http://dx.doi.org/10.1016/j.aei.2023.102024.

[32] D. Kowald, S. Scher, V. Pammer-Schindler, P. Müllner, K. Waxnegger, L. Demelius, A. Fessl, M. Toller, I.G. Mendoza Estrada, I. Simic, V. Sabol, E.E. Veas A. Trügler, R. Kern, T. Nad, S. Kopeinik, Establishing and evaluating trustworthy AI: Overview and research challenges, 2024, arXiv preprint arXiv:2411.09973 https://arxiv.org/abs/2411.09973.

[33] Q. Ren, D. Zhang, M. Li, S. Chen, D. Tian, H. Li, L. Liu, Automatic quality compliance checking in concrete dam construction: Integrating rule syntax parsing and semantic distance, Adv. Eng. Inform. 60 (2024) 102409, http://dx.doi.org/10.1016/j.aei.2024.102409.

[34] T. Beach, J. Yeung, N. Nisbet, Y. Rezgui, Digital approaches to construction compliance checking: Validating the suitability of an ecosystem approach to compliance checking, Adv. Eng. Inform. 59 (2024) 102288, http://dx.doi.org/10.1016/j.aei.2023.102288.

[35] S. Madireddy, L. Gao, Z.U. Din, K. Kim, A. Senouci, Z. Han, Y. Zhang, Large language model-driven code compliance checking in building information modeling, Electronics 14 (11) (2025) 2146, http://dx.doi.org/10.3390/electronics14112146.

[36] W. Mazairac, J. Beetz, BIMQL – An open query language for building information models, Adv. Eng. Inform. 27 (4) (2013) 444–456, http://dx.doi.org/10.1016/j.aei.2013.06.001.

[37] P.N. Gade, K. Svidt, Exploration of practitioner experiences of flexibility and transparency to improve BIM-based model checking systems, J. Inf. Technol. Constr. (ITcon) 26 (2021) 991–1009, http://dx.doi.org/10.36680/j.itcon.2021.055.

[38] B. Zhong, W. He, Z. Huang, P.E.D. Love, J. Tang, H. Luo, A building regulation question answering system: A deep learning methodology, Adv. Eng. Inform. 46 (2020) 101195, http://dx.doi.org/10.1016/j.aei.2020.101195.

[39] H. Wu, B. Zhong, H. Li, P. Love, X. Pan, N. Zhao, Combining computer vision with semantic reasoning for on-site safety management in construction, J. Build. Eng. 42 (2021) 103036, http://dx.doi.org/10.1016/j.jobe.2021.103036.

[40] W. Fang, L. Ma, P.E.D. Love, H. Luo, L. Ding, A. Zhou, Knowledge graph for identifying hazards on construction sites: Integrating computer vision with ontology, Autom. Constr. 119 (2020) 103310, http://dx.doi.org/10.1016/j.autcon.2020.103310.

[41] X. Xue, J. Zhang, Part-of-speech tagging of building codes empowered by deep learning and transformational rules, Adv. Eng. Inform. 47 (2021) 101235, http://dx.doi.org/10.1016/j.aei.2020.101235.

[42] J. Zhang, A logic-based representation and tree-based visualization method for building regulatory requirements, Vis. Eng. 5 (2017) 2, http://dx.doi.org/10.1186/s40327-017-0043-4.

[43] J. Zhang, N.M. El-Gohary, Semantic-based logic representation and reasoning for automated regulatory compliance checking, Autom. Constr. 73 (2017) 45–57, http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000583.

[44] B. Zhong, C. Gan, H. Luo, X. Xing, Ontology-based framework for building environmental monitoring and compliance checking under BIM environment, Build. Environ. (ISSN: 0360-1323) 141 (2018) 127–142, http://dx.doi.org/10.1016/j.buildenv.2018.05.046.

[45] H. Boley, A. Paschke, O. Shafiq, RuleML 1.0: The overarching specification of web rules, semantic web rules (RuleML 2010), in: Lecture Notes in Computer Science, 2010, pp. 162–178, http://dx.doi.org/10.1007/978-3-642-16289-3_15.

[46] C. Preidel, A. Borrmann, Automated code compliance checking based on a visual language and building information modeling, in: Proceedings of the 32nd ISARC, 2015, http://dx.doi.org/10.22260/ISARC2015/0033.

[47] M. Häußler, S. Esser, A. Borrmann, Code compliance checking of railway designs by integrating BIM, BPMN and DMN, Autom. Constr. 121 (2021) 103427, http://dx.doi.org/10.1016/j.autcon.2020.103427.

[48] D. Kampelopoulos, A. Tsanousa, S. Vrochidis, I. Kompatsiaris, A review of LLMs and their applications in the architecture, engineering and construction industry, Artif. Intell. Rev. 58 (2025) 250, http://dx.doi.org/10.1007/s10462-025-11241-7.

[49] C. Sydora, E. Stroulia, Rule-based compliance checking and generative design for building interiors using BIM, Autom. Constr. 120 (2020) 103368, http://dx.doi.org/10.1016/j.autcon.2020.103368.

[50] Y. Zou, Y. Wu, B.H.W. Guo, E. Papadonikolaki, J. Dimyadi, S.N. Hung, Investigating the New Zealand off-site manufacturing industry's readiness for automated compliance checking, J. Constr. Eng. Manag. 148 (12) (2022) 04022148, http://dx.doi.org/10.1061/(ASCE)CO.1943-7862.0002384.

[51] J. Dimyadi, R. Amor, Automated building code compliance checking – where is it at? in: Proceedings of the 19th International CIB World Building Congress, 2013, http://dx.doi.org/10.13140/2.1.4920.4161.

[52] P. Pauwels, T. Mendes de Farias, C. Zhang, A. Roxin, J. Beetz, J. De Roo, C. Nicolle, A performance benchmark over semantic rule checking approaches in construction industry, Adv. Eng. Inform. 33 (2017) 68–88, http://dx.doi.org/10.1016/j.aei.2017.05.001.

[53] S. Fuchs, R. Amor, Natural language processing for building code interpretation: A systematic review, in: Proceedings of the 38th International Conference of CIB W78, Luxembourg, 13–15, 2021, pp. 294–303, https://itc.scix.net/paper/w78-2021-paper-030.

[54] S. Li, Z. Jiang, Z. Xu, BIM-based model checking: A scientometric analysis and critical review, Appl. Sci. 15 (1) (2025) 49, http://dx.doi.org/10.3390/app15010049.

[55] M.L. Rethlefsen, S. Kirtley, S. Waffenschmidt, A.P. Ayala, D. Moher, M.J. Page, J.B. Koffel, PRISMA-S Group, PRISMA-s: An extension to the PRISMA statement for reporting literature searches in systematic reviews, Syst. Rev. 10 (2021) 39, http://dx.doi.org/10.1186/s13643-020-01542-z.

[56] N. Reimers, I. Gurevych, Sentence-BERT: Sentence embeddings using siamese BERT networks, in: Proc. 2019 Conference on Empirical Methods in Natural Language Processing, EMNLP, Association for Computational Linguistics, 2019, http://dx.doi.org/10.18653/v1/D19-1410.

[57] A. Singh, M. D'Arcy, A. Cohan, D. Downey, S. Feldman, SciRepEval: A multi-format benchmark for scientific document representations (introducing SPECTER2), 2022, http://dx.doi.org/10.48550/arXiv.2211.13308, arXiv preprint.

[58] M. Ostendorff, N. Rethmeier, I. Augenstein, B. Gipp, G. Rehm, Neighborhood contrastive learning for scientific document representations with citation embeddings, in: Proc. 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP, Association for Computational Linguistics, 2022, http://dx.doi.org/10.18653/v1/2022.emnlp-main.802.

[59] W. Wang, F. Wei, L. Dong, H. Bao, N. Yang, M. Zhou, MiniLM: Deep self-attention distillation for task-agnostic compression of pre-trained transformers, 2020, http://dx.doi.org/10.48550/arXiv.2002.10957, arXiv preprint.

[60] C.C. Aggarwal, A. Hinneburg, D.A. Keim, On the surprising behavior of distance metrics in high dimensional space, in: Database Theory — ICDT 2001, Lecture Notes in Computer Science, vol. 1973, Springer, Berlin, Heidelberg, 2001, pp. 420–434, http://dx.doi.org/10.1007/3-540-44503-X_27.

[61] J. Eisner, M. Gallé, J. Heinz, A. Quattoni, G. Rabusseau (Eds.), Proceedings of the Workshop on Deep Learning and Formal Languages: Building Bridges, Association for Computational Linguistics, Florence, Italy, 2019, https://aclanthology.org/W19-3900/.

[62] K. Zhou, K. Ethayarajh, D. Card, D. Jurafsky, Problems with cosine as a measure of embedding similarity for high frequency words, 2022, arXiv preprint arXiv:2205.05092 https://arxiv.org/abs/2205.05092.

[63] R. Zhang, N. El-Gohary, Natural language generation and deep learning for intelligent building codes, Adv. Eng. Inform. 52 (2022) 101557, http://dx.doi.org/10.1016/j.aei.2022.101557.

[64] S. Zentgraf, P. Hagedorn, M. König, Multi-requirements ontology engineering for automated processing of document-based building codes to linked building data properties, IOP conference, in: Series: Earth and Environmental Science, vol. 1101, 2022, 092007, http://dx.doi.org/10.1088/1755-1315/1101/9/092007.

[65] Z. Zheng, Y.-C. Zhou, X.-Z. Lu, J.-R. Lin, Knowledge-informed semantic alignment and rule interpretation for automated compliance checking, Autom. Constr. 142 (2022) 104524, http://dx.doi.org/10.1016/j.autcon.2022.104524.

[66] D. Lehoczky, D. García, P.N. Gade, Adapting the 3S-model for investigating trust in AEC solutions in the danish construction industry, J. Inf. Technol. Constr. (ITcon) 27 (2022) 991–1009, http://dx.doi.org/10.36680/j.itcon.2022.048.

[67] J.-K. Lee, K. Cho, H. Choi, S. Choi, S. Kim, S.H. Cha, High-level implementable methods for automated building code compliance checking, Dev. the Built Environ. 15 (2023) 100174, http://dx.doi.org/10.1016/j.dibe.2023.100174.

[68] J. Wu, X. Xue, J. Zhang, Invariant signature, logic reasoning, and semantic natural language processing (NLP)-based automated building code compliance checking (i-SNACC) framework, ITcon 28 (2023) http://dx.doi.org/10.36680/j.itcon.2023.001.

[69] Y. Zhou, Z. Zheng, J. Lin, X. Lu, Integrating NLP and context-free grammar for complex rule interpretation towards automated compliance checking, Comput. Ind. 142 (2022) 103746, http://dx.doi.org/10.1016/j.compind.2022.103746.

[70] R. Zhang, N. El-Gohary, Hierarchical representation and deep learning-based method for automatically transforming textual building codes into semantic computable requirements, J. Comput. Civ. Eng. 36 (5) (2022) 04022022, http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0001014.

[71] S. Fuchs, M. Witbrock, J. Dimyadi, R. Amor, Neural semantic parsing of building regulations for compliance checking, in: IOP Conference Series: Earth and Environmental Science, vol. 1101, 2022, 092022, http://dx.doi.org/10.1088/1755-1315/1101/9/092022.

[72] S. Fuchs, J. Dimyadi, M. Witbrock, R. Amor, Intermediate representations to improve the semantic parsing of building regulations, Adv. Eng. Inform. 62 (part B) (2024) 102735, http://dx.doi.org/10.1016/j.aei.2024.102735.

[73] T. Beach, J. Yeung, A. Ghoroghi, Y. Rezgui, Moving automated compliance checking to the operational phase of the building life-cycle: analysis and feasibility study in the UK, Int. J. Constr. Manag. 25 (7) (2024) 840–849, http://dx.doi.org/10.1080/15623599.2024.2366727.

[74] L. Jia, M. Chen, C. Chen, Y. Jin, A semantic web and IFC-based framework for automated BIM compliance checking, Buildings 15 (2025) 2633, http://dx.doi.org/10.3390/buildings15152633.

[75] Z. Ma, H. Zhu, X. Xiang, Ž. Turk, R. Klinc, Automatic compliance checking of BIM models against quality standards based on ontology technology, Autom. Constr. 166 (2024) 105656, http://dx.doi.org/10.1016/j.autcon.2024.105656.

[76] P. Hagedorn, J. Fauth, S. Zentgraf, S. Seiß, M. König, I. Brilakis, OntoBPR: An ontology-based framework for performing building permit reviews using standardized information containers, Adv. Eng. Inform. 66 (2025) 103369, http://dx.doi.org/10.1016/j.aei.2025.103369.

[77] S. Zhu, J. Zhou, L. Cheng, X. Fu, Y. Wang, K. Dai, Research on a BIM model quality compliance checking method based on a knowledge graph, J. Comput. Civ. Eng. 39 (1) (2025) 04024049, http://dx.doi.org/10.1061/JCCEE5.CPENG-5950.

[78] E. Makisha, Features of regulation document translation into a machine-readable format within the verification of building information models, CivilEng 4 (2) (2023) 373–390, http://dx.doi.org/10.3390/civileng4020022.

[79] P. Zech, P. Burger, S. Hammes, D. Geisler-Moroder, R. Breu, Bimreason: Validating BIM model correctness, Bauphysik 46 (6) (2024) 332–339, http://dx.doi.org/10.1002/bapi.202400035.

[80] R. Kruiper, B. Kumar, R. Watson, F. Sadeghineko, A. Gray, I. Konstas, A platform-based natural language processing-driven strategy for digitalising regulatory compliance processes for the built environment, Adv. Eng. Inform. 62 (2024) 102653, http://dx.doi.org/10.1016/j.aei.2024.102653.

[81] A. Karmakar, V.S.K. Delhi, Semantic BIM enrichment using a hybrid ML and rule-based framework for automated tenement compliance checking, Autom. Constr. 177 (2025) 106369, http://dx.doi.org/10.1016/j.autcon.2025.106369.

[82] W. Goh, J. Jang, S.I. Park, B.-H. Choi, H. Lim, G. Zi, Automated compliance checking system for structural design codes in a BIM environment, KSCE J. Civ. Eng. 28 (10) (2024) 4175–4189, http://dx.doi.org/10.1007/s12205-024-1121-5.

[83] Z. Kabzhan, A. Shakhnovich, S. Gorshkov, Y. Yemenov, F. Gorshkov, N. Shogelova, Semantic and ontology-based analysis of regulatory documents for construction industry digitalization, Front. Built Environ. 11 (2025) 1575913, http://dx.doi.org/10.3389/fbuil.2025.1575913.

[84] J. Wu, J. Zhang, Model validation using invariant signatures and logic-based inference for automated building code compliance checking, J. Comput. Civ. Eng. 36 (3) 2022, http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0001002.

[85] X. Xue, J. Wu, J. Zhang, Semiautomated generation of logic rules for tabular information in building codes to support automated code compliance checking, J. Comput. Civ. Eng. 36 (1) 2022, http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0001000.

[86] M. Yang, Q. Zhao, L. Zhu, H. Meng, K. Chen, Z. Li, X. Hei, Semi-automatic representation of design code based on knowledge graph for automated compliance checking, Comput. Ind. 150 (2023) 103945, http://dx.doi.org/10.1016/j.compind.2023.103945.

[87] F. Yang, J. Zhang, Prompt-based automation of building code information transformation for compliance checking, Autom. Constr. 168 (2024) 105817, http://dx.doi.org/10.1016/j.autcon.2024.105817.

[88] P. Patlakas, I. Christovasilis, L. Riparbelli, F.K.T. Cheung, E. Vakaj, Semantic web-based automated compliance checking with integration of finite element analysis, Adv. Eng. Inform. 61 (2024) 102448, http://dx.doi.org/10.1016/j.aei.2024.102448.

[89] H. Li, R. Yang, S. Xu, Y. Xiao, H. Zhao, Intelligent checking method for construction schemes via fusion of knowledge graph and large language models, Buildings 14 (2024) 2502, http://dx.doi.org/10.3390/buildings14082502.
[90] X. Wang, N. El-Gohary, Deep learning-based relation extraction and knowledge graph-based representation of construction safety requirements, Autom. Constr. 147 (2023) 104696, http://dx.doi.org/10.1016/j.autcon.2022.104696.
[91] Y. Chen, H. Jiang, Optimizing automated compliance checking with ontology-enhanced natural language processing: Case in the fire safety domain, J. Environ. Manag. 371 (2024) 123320, https://api.semanticscholar.org/CorpusID:274092291.
[92] J. Zhang, How can ChatGPT help in automated building code compliance checking? in: Proceedings of the 40th International Symposium on Automation and Robotics in Construction, ISARC, Chennai, India, 2023, pp. 63–70, http://dx.doi.org/10.22260/ISARC2023/0011.
[93] J.W.L. Shi, W. Solihin, J.K.W. Yeoh, Fine-tuning a large language model for automated code compliance of building regulations, Adv. Eng. Inform. 68 (Part B) (2025) 103676, http://dx.doi.org/10.1016/j.aei.2025.103676.
[94] K.W. Johansen, J. Teizer, C. Schultz, Automated rule-based safety inspection and compliance checking of temporary guardrail systems in construction, Autom. Constr. 168 (2024) 105849, http://dx.doi.org/10.1016/j.autcon.2024.105849.
[95] I. Huitzil, M. Schorlemmer, N. Osman, P. Garcia, J. Coll, X. Coll, Towards automated compliance checking of building regulations: smartNorms4BIM, in: Artificial Intelligence Research and Development, IOS Press E-books, 2022, pp. 95–104, http://dx.doi.org/10.3233/FAIA220322.
[96] A.S. Ismail, K.N. Ali, N.A. Iahad, M.A. Kassem, N.T. Al-Ashwal, BIM-based automated code compliance checking system in Malaysian fire safety regulations: A user-friendly approach, Buildings 13 (2023) 1404, http://dx.doi.org/10.3390/buildings13061404.
[97] Y. Li, M. Yang, Q. Zhao, Z. Li, Z. Ma, Y. Liu, X. Hei, A semantic representation method of building codes applied to compliance checking, Mathematics 11 (2023) 2552, http://dx.doi.org/10.3390/math11112552.
[98] O. Doukari, D. Greenwood, K. Rogage, M. Kassem, Object-centred automated compliance checking: A novel bottom-up approach, J. Inf. Technol. Constr. (ITcon) 27 (2022) 335–362, http://dx.doi.org/10.36680/j.itcon.2022.017.
[99] Y. Zhou, J. She, Y. Huang, L. Li, L. Zhang, J. Zhang, A design for safety (DFS) semantic framework development based on natural language processing (NLP) for automated compliance checking using BIM: The case of China, Buildings 12 (2022) 780, http://dx.doi.org/10.3390/buildings12060780.
[100] G. Recski, E. Iklódi, B. Lellmann, Á. Kovács, A. Hanbury, BRISE-plandok: A german legal corpus of building regulations, Lang. Resour. Eval. 59 (2025) 1043–1082, http://dx.doi.org/10.1007/s10579-024-09747-7.
[101] R. Kruiper, I. Konstas, A.J.G. Gray, F. Sadeghineko, R. Watson, B. Kumar, Taking stock: A linked data inventory of compliance checking terms derived from building regulations, in: W. Terkaj, M. Poveda-Villalón, P. Pauwels (Eds.), in: Proceedings of the 11th Linked Data in Architecture and Construction Workshop (LDAC2023), CEUR Workshop Proceedings, Vol. 3633, Matera, Italy, 2023, pp. 222–237, https://linkedbuildingdata.net/ldac2023/files/papers/papers/LDAC2023_paper_4692.pdf.
[102] H. Ying, R. Sacks, From automatic to autonomous: a large language model-driven approach for generic building compliance checking, 2024, EasyChair Preprint 15364 https://easychair.org/publications/preprint/k2PJ.
[103] J. Anim, L. Robaldo, A.Z. Wyner, A SHACL-based approach for enhancing automated compliance checking with RDF data, Information 15 (2024) 759, http://dx.doi.org/10.3390/info15120759.
[104] V. Schuk, M.E. Pombo Jiménez, U. Martin, Technical specifications to meet the requirements of an automatic code compliance checking tool and current developments in infrastructure construction, Results Eng. 16 (2022) 100650, http://dx.doi.org/10.1016/j.rineng.2022.100650.
[105] T. Bloch, A. Borrmann, P. Pauwels, Graph-based learning for automated code checking – exploring the application of graph neural networks for design review, Adv. Eng. Inform. 58 (2023) 102137, http://dx.doi.org/10.1016/j.aei.2023.102137.
[106] L. Robaldo, F. Pacenza, J. Zangari, R. Calegari, F. Calimeri, G. Siragusa, Efficient compliance checking of RDF data, J. Logic Comput. 33 (2023) 1776, http://dx.doi.org/10.1093/logcom/exad034.
[107] X. Liu, H. Li, X. Zhu, A GPT-based method of automated compliance checking through prompt engineering, in: Proceedings of the 30th European Group for Intelligent Computing in Engineering (EG-ICE 2023): Towards Sustainable, Smart and Resilient Buildings, Infrastructures and Cities, University College London, UK, 2023, pp. 4–7, https://orca.cardiff.ac.uk/id/eprint/159694.
[108] Z. Wu, Z. Wang, J.C.P. Cheng, H.H.L. Kwok, A knowledge-informed optimization framework for performance-based generative design of sustainable buildings, Appl. Energy 367 (2024) 123318, http://dx.doi.org/10.1016/j.apenergy.2024.123318.
[109] J. Wu, S. Nousias, A. Borrmann, Design healing framework for automated code compliance, Autom. Constr. 171 (2025) 106004, http://dx.doi.org/10.1016/j.autcon.2025.106004.
[110] D. Chen, L. Chen, Y. Zhang, S. Lin, M. Ye, S. Sølvsten, Automated fire risk assessment and mitigation in building blueprints using computer vision and deep generative models, Adv. Eng. Inform. 62 (part A) (2024) 102614, http://dx.doi.org/10.1016/j.aei.2024.102614.
[111] H. Gao, B. Zhong, H. Luo, W. Chen, Computational geometric approach for BIM semantic enrichment to support automated underground garage compliance checking, J. Constr. Eng. Manag. 148 (1) 2021, http://dx.doi.org/10.1061/(ASCE)CO.1943-7862.0002230.
[112] A. Karmakar, C. Patel, V.S.K. Delhi, From unstructured data to knowledge graphs: An application for compliance checking problem, in: Proceedings of the 41st International Symposium on Automation and Robotics in Construction, ISARC 2024, Lille, France, 2024, pp. 863–871, http://dx.doi.org/10.22260/ISARC2024/0112.