2026 European Conference on Computing in Construction
Corfu, Greece
July 12–15, 2026

IDS-Constrained Text-to-Graph Querying for IFC Models

Ivan Smirnov[1], Tanya Bloch[2], Fulvio Re Cecconi[1],
Sebastiano Maltese[3], Lavinia Chiara Tagliabue[4], Silvia Meschini[4] and Shabtai Isaac[5]
[1]Politecnico di Milano, Milan, Italy
[2]Technion – Israel Institute of Technology, Haifa, Israel
[3]University of Applied Sciences and Arts of Southern Switzerland, Mendrisio, Switzerland
[4]University of Turin, Turin, Italy
[5]Ben-Gurion University of the Negev, Beersheba, Israel

Abstract

Accessing Building Information Model (BIM) data encoded in Industry Foundation Classes (IFC) remains difficult for non-expert stakeholders due to heterogeneous property sets and nested relationships. This paper presents a constrained text-to-graph querying framework that transforms IFC data into a labeled property graph, queried via Cypher, while project-specific Information Delivery Specification (IDS) requirements define admissible entity labels, relationships, and property keys. A regular-expression grammar restricts Large Language Model (LLM) generation to executable, schema-compliant queries. Evaluation shows grammar constraints eliminate schema hallucinations, achieving 100% Semantic Compliance and improved execution accuracy, with competitive performance from smaller open-weights models.

Introduction

BIM workflows increasingly presume that building information can be accessed on demand to support coordination, compliance checking, and decision-making. In practice, however, obtaining a first correct and verifiable answer from an IFC model remains costly (Hellin et al., 2025). This cost extends beyond computation to include the time and expertise required to translate an information need into a formal query, execute it on an IFC-derived representation, and validate the results against both the model content and project-specific modelling conventions. This initial problem typically involves multiple iterations of identifying relevant entities and relations, selecting appropriate properties and value types, and refining queries until defensible results are produced. The burden is most evident for information needs that exceed direct attribute lookups and require combinations of property constraints and topological reasoning. Common quality and compliance checks, such as isolating spatially contained elements or verifying classification and completeness requirements, remain technically demanding, as they require detailed knowledge of IFC structures and querying mechanisms. Prior research consistently identifies an intent-to-query gap in BIM information access and proposes abstractions and natural-language interfaces to mitigate direct interaction with raw IFC schemas (Mazairac and Beetz, 2013; Yin et al., 2023a; Hellin et al., 2025). However, query verbosity and representational overhead remain pragmatic obstacles, particularly where relevant information is accessible only through long navigation paths and reified relationship structures. IFC, as an open and vendor-neutral standard (International Organization for Standardization, 2024), encodes semantics indirectly, leading to complex multi-hop queries across relationship entities and nested value representations. Linked Data approaches, including ifcOWL and higher-level ontologies such as the Building Topology Ontology (BOT) and BimSPARQL, improve expressiveness and topology-centric querying (Beetz et al., 2009; Rasmussen et al., 2020; Zhang et al., 2018). Nevertheless, semantic variance across projects, arising from custom property sets, naming conventions, heterogeneous datatypes, and tool-specific export practices, limits query robustness and reuse (Yu et al., 2023). IDS addresses this by formalising project-specific information requirements and constraints in a machine-interpretable manner (buildingSMART International, 2024).

Recent advances in LLMs demonstrate strong capabilities in natural-language-to-structure translation (Jiang et al., 2023), yet their probabilistic nature poses reliability risks when applied to deterministic, schema-bound engineering data. Hallucinations and invalid query generation remain critical concerns (Huang et al., 2023). Building on recent BIM research that combines LLMs with graph-based grounding (Iranmanesh et al., 2025), this paper proposes a reliability-oriented framework that translates natural-language questions into constrained graph queries over IFC-derived representations, informed by IDS constraints. A key objective is to enable the framework to run privately on local machines, preserving data confidentiality and substantially reducing the energy and operational costs associated with reliance on commercial, cloud-based LLM services. By combining graph abstractions, explicit project constraints, and constrained decoding, the approach aims to improve the correctness and verifiability of results for topology- and constraint-dependent BIM checks.

Background and Related Work

Querying BIM models

BIM models integrate geometry, spatial relations, material properties, quantities, and semantics in one digital structure. In practice, this richness creates large datasets that are difficult to interrogate without specialized tools. Most exchanges rely on IFC (currently version 4) (build-

ingSMART International, 2024), which provides interoperability but also substantial complexity due to broad entity vocabularies, heterogeneous property sets, and deeply nested relations (Borrmann et al., 2018). As repositories grow, efficient BIM search becomes increasingly important (Khademi and Behan, 2017). Early efforts introduced domain-specific query languages such as BIMQL (Mazairac and Beetz, 2013) and QL4BIM (Daum and Borrmann, 2014). BIMQL focuses on class- and attribute-based filtering, whereas QL4BIM emphasizes spatial and topological queries. Both approaches improve expressiveness but still require users to understand formal query syntax and IFC internals, including indirect relationship encoding. Database-based approaches have also been explored. SQL is mature and efficient, but less suited to IFC-style hierarchical and multi-hop relations. SPARQL over ifcOWL maps IFC into RDF triples and supports linked-data querying (Zhang et al., 2018), yet this representation is often verbose and large (Pauwels et al., 2017). As an alternative, labeled property graphs (LPGs) represent relations more directly and support efficient traversal in systems such as Neo4j (Neo4j, 2026), queried with Cypher (Zhu et al., 2025). The RDF–LPG choice remains use-case dependent, but LPGs are often preferred for storage and query efficiency (Zhu et al., 2023). Prior work shows the value of LPG-based BIM querying. Ismail et al. (2017) demonstrated IFC-to-LPG transformation for graph analyses (e.g. emergency routes and model comparison), while later studies reported benefits for change tracking (Esser et al., 2022), semantic enrichment (Wang et al., 2022), and code compliance (Bloch et al., 2023). With recent LLM advances, graph representations are increasingly relevant as a grounding layer for natural-language BIM interaction and hallucination reduction.

LLM based BIM query

Natural-language BIM querying lowers the barrier for non-expert users by avoiding direct interaction with formal query syntax. However, LLMs alone do not reliably query IFC data: without grounding, they hallucinate entities, properties, and relations. Du et al. (2026) identify three recurring causes: long prompt context, limited global spatial understanding, and cumulative errors in multi-agent pipelines. The spatial reasoning gap is particularly critical for BIM querying. Accordingly, prior work introduces external constraints. Ontology-based approaches translate natural language into SPARQL over RDF/OWL models, as in Yin et al. (2023b). Tool-augmented pipelines instead couple LLM planning with deterministic BIM operations; for example, Hellin et al. (2025) use a ReAct-style workflow with Python tools over IFC, reporting good results for direct queries but weaker performance for indirect and relational questions. As also discussed by Du et al. (2026), graph representations can improve structural grounding for spatial and topological reasoning. Retrieval-augmented generation (RAG) provides another grounding strategy by injecting retrieved context at inference time (Gao et al., 2023). In BIM, this can supply IFC documentation and project-specific metadata (Iranmanesh et al., 2025). Yet text-centric RAG does not fully capture relational model structure. Graph-RAG improves this by retrieving connected nodes and edges, but still underperforms on complex multi-hop queries with multiple entity references and ambiguity (Iranmanesh et al., 2025). Formal project specifications offer a complementary constraint layer. IDS (buildingSMART International, 2024) defines expected IFC entities, properties, value domains, and vocabularies in machine-readable XML, primarily for automated compliance checking. In query generation, IDS can constrain allowable query patterns across IFC-, RDF-, or LPG-based representations, grounding LLM output in explicit project rules. This study therefore examines whether project-specific IDS constraints reduce hallucinations in LLM-mediated BIM querying, a direction that remains largely unexplored.

Appropriate use cases and study scope

From a theoretical perspective, model checking activities can be classified into three categories (Solihin and Eastman, 2015; Hjelseth and Nisbet, 2016). Simple checks involve the verification of the existence of specific property sets, the presence of assigned values, and the correctness of those values. Elaborated checks address design, spatial, or technical characteristics that are not explicitly stored as IFC properties but can be derived through relatively straightforward algorithms or mathematical relationships. A representative example is the verification of regulatory requirements governing the minimum ratio between window surface area and room floor area, as prescribed by Italian building codes. Complex checks, by contrast, focus on performance-based assessments and require advanced simulations, such as those used for evaluating building energy performance.

For simple checks, established rule-based approaches (e.g. IDS frameworks and commercial tools like Solibri) are likely to remain more efficient and reliable than LLM-based solutions. In this context, the primary advantage of LLMs lies in their usability, as they lower the technical barrier for non-specialist users. The suitability of LLMs for elaborated checks is clearer, however, concerns remain regarding computational accuracy and the risk of hallucinated outputs (Huang et al., 2023). In these cases, LLMs may be a useful interface between the complexity of rule-based checking software and end users, providing the ability to perform elaborated checks without needing to learn the software's query grammar. With respect to complex checks, neither traditional rule-based tools nor LLM-based approaches currently offer adequate capabilities.

Beyond model checking, greater value may be found in leveraging LLMs as intuitive interfaces for querying BIM data (Yin et al., 2023b). Their accessibility enables meaningful interaction with BIM models by non-technical stakeholders, including building occupants, maintenance personnel, municipal officers involved in digital permit-

ting, and senior decision-makers within organizations. This work focuses on elaborated checks, where the key challenge is reliably translating natural-language intent into verifiable, executable queries over IFC-derived data.

Research Questions

To structure the study, three research questions are formulated, each addressing a key design choice of the proposed pipeline:

RQ1 (text-to-query over direct QA): Can LLM-mediated text-to-graph querying produce correct results for IFC-based information needs where Direct QA over serialized data fails?

RQ2 (graph representation): How does representing IFC as a graph affect the correctness and robustness of answers generated from queries over the model?

RQ3 (IDS as a constraint layer): Does integrating IDS improve the semantic correctness of LLM-generated Cypher queries over IFC-derived graphs?

Together, these questions assess whether a constrained graph query workflow can provide predictable, auditable access to BIM information for non-expert stakeholders, while remaining practical for real project models.

Proposed Method

System overview

A Constrained Natural Language to Cypher (NL2Cypher) framework is proposed to ground LLMs in project-specific constraints defined by the IDS. An IFC file is transformed into a Neo4j (Neo4j, 2026) LPG and an IDS file is compiled into a formal grammar. This grammar restricts the LLM's decoding process, ensuring that generated Cypher queries are syntactically valid and semantically compliant with the project schema.

The architecture consists of three primary layers:

a) The data ingestion layer transforms the IFC model into a flattened property graph and scans the model for available vocabulary.

b) The constraint layer parses the IDS to extract enforced constraints (entities, properties, enums) and merges them with the model vocabulary to build a strictly typed schema.

c) The generation layer uses constrained decoding to guide the LLM, masking invalid tokens at each generation step based on a regular expression derived from the schema.

IFC-to-graph mapping

To enable efficient graph transit, the system simplifies the complex IFC schema into a LPG. The mapping process involves three transformation rules. First, every `IfcProduct` becomes a graph node. The node label corresponds to the IFC class (e.g., `IfcWall`). Then, the relationships hosted in `IfcPropertySet` entities linked via `IfcRelDefinesByProperties` are flattened and embedded directly onto the entity node as key-value pairs. Finally, the complex relationship entities (e.g., `IfcRelAggregates`, `IfcRelContainedInSpatialStructure`) are converted into directed edges. The relationships most commonly required for property- and topology-based queries are prioritized: CONTAINS (from `IfcRelContainedInSpatialStructure`), DECOMPOSES (from `IfcRelAggregates`), and HAS_MATERIAL (from `IfcRelAssociatesMaterial`). Relationship types that serve different querying needs—such as `IfcRelVoidsElement` and `IfcRelFillsElement` (opening and infill associations), `IfcRelConnectsPathElements` (wall connectivity), and `IfcRelSpaceBoundary` (thermal envelope geometry)—are not mapped in the current implementation. The mapping is extensible: supporting additional relationship types requires adding the corresponding extraction logic and registering new edge labels in the grammar.

IDS parsing and constraint compilation

The core innovation of this method is the translation of the IDS "semantic contract" into a generative grammar. Solely relying on the IDS can be insufficient if the model contains valid data not covered by the specification. At the same time relying only on the model allows the LLM to hallucinate on dirty data. Therefore a Hybrid schema merging two sources is constructed.

Enforced Vocabulary from IDS defines strict constraints. The system parses the IDS XML structure, extracting entities from `applicability` clauses and required properties from `requirements` clauses. It captures domain-specific constraints defined via XML Schema Definition (XSD) restrictions (e.g., `xs:enumeration`), ensuring that sensitive properties (e.g., `FireRating`) are strictly limited to their defined enums.

Available Vocabulary from IFC is extracted by scanning the model file for existing labels and properties. Conflicts are resolved by giving precedence to the IDS. Properties are categorized into two tiers: **strict** (constrained to the IDS-defined enums) and **open** (inferred from the model with data types like Boolean, Numeric or String). This ensures regulatory compliance without rendering valid non-standard model data unqueryable.

This hybrid vocabulary is then compiled into a Regular Expression (Regex) that guides the LLM's decoding process via Outlines library (dottxt-ai, 2026).

Query generation

Instead of fine-tuning the model or relying on "soft" prompt engineering, constrained decoding is utilized. The hybrid schema is translated into a regex pattern that defines the set of all possible valid Cypher queries for that specific building. For a schema with entities E and properties P, a regular expression $r = \texttt{build\_cypher\_regex}(E, P)$ is compiled to act as a grammar for valid Cypher. r is assembled from (i) an entity alternation group E_{alt} (e.g., `(IfcDoor|IfcWall)`), (ii) a property alternation group P_{alt} (e.g., `(FireRating|Height|Name)`), and (iii) a small set of admissible literals and operators. Property access is restricted to the bound match vari-

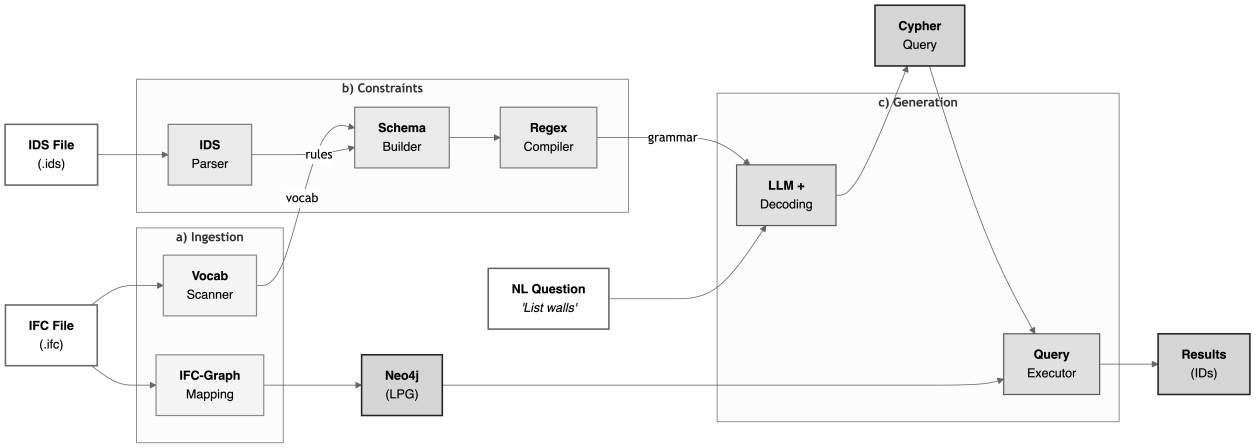

Figure 1: System overview of the NL2Cypher framework. The three layers correspond to (a) data ingestion, (b) constraint compilation, and (c) grammar-constrained query generation.

able (e.g., v.`Name`) with optional type conversions (e.g., `toInteger`(v.`FireRating`)). The grammar supports semantic string matching (e.g., $CONTAINS$) and null-safety checks ($ISNOTNULL$) which are essential for handling natural language ambiguity. The `WHERE` clause is optional and bounded to a fixed maximum number of boolean conditions to avoid overly long or incomplete queries.

$$
\begin{aligned}
r &= r_{\text{MATCH}}\, r_{\text{REL}}^{?}\, r_{\text{WHERE}}^{?}\, r_{\text{RETURN}}, \\
r_{\text{MATCH}} &:= \texttt{MATCH}\,(v : E_{\text{alt}}), \\
r_{\text{REL}} &:= -[: R_{\text{alt}}] \rightarrow (v_2 : E_{\text{alt}}), \\
r_{\text{WHERE}} &:= \texttt{WHERE}\; c_1\; \big((\texttt{AND}|\texttt{OR})\; c_2\big)^{\leq 4}, \\
c_i &:= \big(f^{?}(v.P_{\text{alt}})\ \text{op}\ \text{val}\big) \mid \big(v.P_{\text{alt}}\ \texttt{CONTAINS}\ \text{str}\big), \\
f &\in \{\texttt{toInteger}, \texttt{toFloat}, \texttt{toString}\}.
\end{aligned}
\tag{1}
$$

Note that R_{alt} represents the relationship alternation group (e.g., (CONTAINS|HAS_MATERIAL)). Furthermore, the return clause r_{RETURN} is constrained to return the node variable v. This is strictly enforced to guarantee that the execution engine retrieves the element's `GlobalId` for evaluation against the Gold Standard, preventing the generation of queries that return unidentifiable scalar values.

During inference, the constrained decoder only allows tokens that keep the partial output compatible with r (invalid next tokens are masked by setting their logits to $-\infty$). As a result, generated queries are (i) executable (high SVR - Syntactic Validity Rate) and (ii) limited to entity labels and property names present in the hybrid schema (high SCR - Schema Coverage/Compliance Rate).

Experimental Design

Baselines and datasets for demonstration and validation

To evaluate the efficacy of IDS grounding and graph-based querying, four experimental settings are compared in a 2×2 matrix of Direct Question Answering (QA) versus Text-to-Cypher, crossed with baseline versus IDS-constrained configurations (Table 1.)

Table 1: Experiment settings.

	Direct QA	Text-to-Cypher
Baseline	**S1** Naive QA **Input:** IFC text dump **Output:** JSON `GlobalIds`	**S3** Soft Cypher **Input:** graph schema **Output:** Cypher → IDs
With IDS	**S2** Grounded QA **Input:** IFC + IDS rules **Output:** JSON `GlobalIds`	**S4** Strict Cypher (proposed) **Input:** hybrid schema (IDS + model) **Output:** Cypher → IDs

Settings 1&2 simulate a RAG approach where the model reasons over raw text. To ensure comparability, the LLM is instructed to output a JSON list of `GlobalId` strings rather than the natural language. Settings 3&4 generate executable Cypher code. Setting 3 represents the current state-of-the-art in Text-to-Cypher (prompting), while Setting 4 represents our proposed method.

Dataset and Preprocessing

A sample project from Autodesk (Autodesk, 2026), exported as an IFC model, and a corresponding IDS requirement file are utilized. For the Direct QA settings (1, 2), the IFC model is serialized into a condensed JSON format. For the Graph Settings (3, 4), the model is loaded into a Neo4j (Neo4j, 2026) instance using the mapping de-

scribed above.

The IDS file defines seven specifications that constrain naming conventions, required properties, and material assignments for the principal IFC entities (Table 2). Properties such as `FireRating` are limited to enumerated values (0, 30, 60, 90 minutes), while `LoadBearing` and `IsExternal` are enforced as Boolean indicators. Material constraints require both presence (for structural elements) and specific content (concrete for load-bearing columns). These specifications form the enforced vocabulary layer in the hybrid schema.

Table 2: Summary of IDS specifications used in the evaluation.

Applicable Entity	Required Properties	Value Constraints
`IfcBuildingStorey`	`Name`	Level naming pattern
`IfcDoor`	`Name`, `IsExternal`, `FireRating`	`FireRating` ∈ {0, 30, 60, 90}
`IfcSpace`	`NetFloorArea`, `Reference`	Controlled room-type vocabulary
`IfcWall`	`LoadBearing`, `IsExternal`	Boolean
`IfcWindow`	`Name`	Component naming pattern
Structural†	`Material`	Presence required
`IfcColumn`*	`Material`	Must contain ''Concrete''

*Load-bearing only. †`IfcStair`, `IfcBeam`, `IfcSlab`, `IfcPile`, `IfcRamp`, `IfcColumn`.

Task design and use cases

The evaluation consists of 16 natural-language queries, categorized into three levels of increasing complexity:

- **Level 1 – Class retrieval** (6 queries, Easy): direct entity lookups that test the ability to identify correct IFC classes. Example: *''List all doors''* → `MATCH (n:IfcDoor) RETURN n`.
- **Level 2 – Property filtering** (5 queries, Medium/Hard): queries that add attribute constraints, Boolean checks, numeric comparisons, or null-safety tests. Examples: *''List all external walls''* → `MATCH (n:IfcWall) WHERE n.IsExternal = true RETURN n`; *''List all doors that have a fire rating defined''* → `MATCH (n:IfcDoor) WHERE n.FireRating IS NOT NULL RETURN n`.
- **Level 3 – Topology and relationships** (5 queries, Medium/Hard): multi-hop graph traversals combining relationship patterns with property constraints. Example: *''List spaces on Level 2 larger than 12 m^2''* → `MATCH (s:IfcBuildingStorey)-[:DECOMPOSES]->(n:IfcSpace) WHERE s.Name CONTAINS '02' AND n.GrossFloorArea > 12 RETURN n`.

Each query is paired with a gold-standard Cypher query whose result set defines the ground-truth `GlobalId`s used for evaluation.

Evaluation

Three metrics are employed to evaluate performance. To enable a common comparison across all settings, the primary metric is the accuracy of the returned `GlobalId`s:

- **Execution Accuracy (EA):** Jaccard similarity between the set of `GlobalId`s returned by the system and the ground-truth set.

$$EA = \frac{|R_{\text{gen}} \cap R_{\text{gold}}|}{|R_{\text{gen}} \cup R_{\text{gold}}|} \tag{2}$$

  For Direct QA (Settings 1&2): ground truth is parsed from the JSON output.
  For Text-to-Cypher (Settings 3&4): ground truth is obtained by executing the generated Cypher query against the database.
- **Syntactic Validity Rate (SVR):** for Direct QA, the percentage of responses that are valid, parseable JSON; for Cypher, the percentage of generated queries that execute without syntax errors.
- **Semantic Compliance Rate (SCR):** the percentage of generated schema elements (entity labels, property names) that exist in the hybrid schema (reported for the Cypher-generating settings).

Results and Discussion

Comparative performance

Phase 1: Cloud baselines and data access

To establish a performance ceiling and verify the necessity of structured data access (RQ1), the framework was first evaluated using state-of-the-art cloud models: Google Gemini 2.5 Flash and Gemini 3.0 Flash Preview. These models were tested on the unconstrained settings: Naive QA (S1), Grounded QA (S2), and Soft Cypher (S3).

Table 3 presents the results of the cloud-based evaluation. Both Direct QA settings (S1 and S2) achieved near-zero EA across all models and runs (Gemini 2.5 Flash: 0.09 ± 0.08; Gemini 3.0 Flash: 0.05 ± 0.06). Despite the advanced reasoning capabilities of the Gemini models, the system failed to reliably identify correct elements. The size of the IFC model chosen for tests exceeds 1M tokens both as a .ifc and as a .json dump, so the truncation of the input plays an important role in such a low performance, too. This confirms that without structured access to the underlying model data, LLMs cannot reliably retrieve specific BIM elements. The unique nature of IFC identifiers (22-character encoded strings) prevents probabilistic guessing, and schema knowledge alone (S2) proved insufficient for identifying specific instances.

In contrast, the graph-based approach (S3: Soft Cypher) demonstrated significant potential even without grammar constraints. Gemini 2.5 Flash achieved a mean EA of 0.67 ± 0.03 across three runs, successfully translating natural language into executable Cypher queries for the majority of test cases.

This disparity supports the hypothesis that the ''Text-to-Graph'' abstraction is superior to ''Direct QA'' for engineering tasks. The LLM excels at logic generation (constructing the query pattern) but struggles with direct information retrieval when denied an explicit index.

While Setting 3 (Soft Cypher) performed well, it was not flawless. Gemini 2.5 Flash achieved a mean SVR of 0.83 ± 0.07, meaning that on average roughly 17% of generated

Table 3: Phase 1 Results: Cloud-based Models (Unconstrained). Values reported as mean ± std over n independent runs.

Model	Setting	SVR	SCR	EA
Gemini 2.5 Flash	S1 (Naive)	0.54 ± 0.14	–	0.09 ± 0.08
($n = 3$)	S2 (Grounded)	0.60 ± 0.10	–	0.08 ± 0.07
	S3 (Soft Cypher)	0.83 ± 0.07	0.99 ± 0.01	0.67 ± 0.03
Gemini 3.0 Flash	S1 (Naive)	0.44 ± 0.17	–	0.05 ± 0.06
(Preview) ($n = 3$)	S2 (Grounded)	0.44 ± 0.06	–	0.02 ± 0.04
	S3 (Soft Cypher)	0.67 ± 0.03	0.99 ± 0.02	0.62 ± 0.03

queries were invalid Cypher and could not be executed. Detailed analysis of the error logs revealed two primary failure modes in the ''Soft'' setting:

1. **Hallucinated properties:** the model occasionally queried properties that did not exist in the graph (SCR < 1.0), despite the schema being provided in the system prompt.
2. **Complexity degradation:** performance dropped sharply as query complexity increased. For ''Hard'' queries (involving relationships and multi-hop traversals), the mean EA for Gemini 2.5 Flash dropped from 100% (Easy) to 22% (Hard).

These limitations in the unconstrained "Soft" approach (specifically a non-perfect SVR) motivate the need for the grammar-constrained generation (S4) explored in the subsequent phase of this study.

Phase 2: Efficacy of grammar constraints

To evaluate the proposed grammar-constrained generation (S4), Qwen2.5-Coder was utilized as a state-of-the-art open-weights model family. Unlike the prompt-based approaches in Phase 1, Setting 4 requires direct intervention in the decoding process to enforce adherence to the hybrid schema; this requirement motivates the use of local, open-weights models rather than API-based cloud services.
Table 4 summarizes the performance impact of the constraints. S3 results are reported as mean ± std over multiple independent runs ($n = 5$ for 7B, $n = 9$ for 3B); S4 results are deterministic (greedy decoding with grammar constraints) and therefore reported from a single run.

Table 4: Phase 2 Results: Open-weights models (constrained vs. unconstrained). S3 values reported as mean ± std over n runs; S4 is deterministic (single run).

Model	Setting	SVR	SCR	EA
Qwen2.5-Coder-7B Instruct	S3 ($n = 5$)	0.88 ± 0.04	0.87 ± 0.01	0.34 ± 0.03
	S4 (Strict)	**1.00**	**1.00**	**0.63**
Qwen2.5-Coder-3B Instruct	S3 ($n = 9$)	0.95 ± 0.03	0.94	0.62 ± 0.02
	S4 (Strict)	**1.00**	**1.00**	**0.64**

The effect of grammar constraints varied markedly between the two models. For the 7B model, which achieved only $EA = 0.34 \pm 0.03$ under unconstrained prompting (S3), grammar constraints nearly doubled accuracy to $EA = 0.63$, while also raising SCR from 0.87 ± 0.01 to a perfect 1.00. In this case, constrained decoding compensated for the model's weaker query-generation logic by narrowing the output space to valid schema elements.

The 3B model tells a different story. It already achieved $EA = 0.62 \pm 0.02$ ($n = 9$) under Soft Cypher—comparable to the 7B model's *constrained* result—with low variance across runs. Grammar constraints (S4) brought EA only marginally higher to 0.64, a difference well within the S3 standard deviation. The benefit of S4 for the 3B model was therefore not in accuracy but in *reliability*: SVR improved from 0.95 ± 0.03 to 1.00 and SCR from 0.94 to 1.00, guaranteeing that every generated query is syntactically valid and schema-compliant. This distinction matters in production settings, where even a 5% rate of invalid queries can erode user trust.

These results suggest that grammar constraints serve two distinct roles depending on model capability: for weaker models, they act as an accuracy amplifier by restricting the search space; for stronger models that already generate correct logic, they act as a safety net that eliminates the residual tail of schema hallucinations and syntax errors.

To further verify the effect of grammar constraints (S4) relative to the prompt-based baseline (S3), results were decomposed by query complexity: Level 1 (Class Retrieval), Level 2 (Property Constraints), and Level 3 (Topology/Relationships).

- **Level 1: Class retrieval (easy).** Under Strict Cypher (S4), both the 7B and 3B models achieved perfect performance ($EA = 1.00$, $SCR = 1.00$, $SVR = 1.00$). In the unconstrained S3 setting, the 3B model already performed strongly ($EA = 0.98 \pm 0.06$), while the 7B model showed more variability ($EA = 0.30 \pm 0.07$). The transition to S4 eliminated all remaining instances of invalid entity labeling.
- **Level 2: Property value retrieval (medium).** The 3B model showed high resilience in this category already under Soft Cypher ($EA = 0.79 \pm 0.02$). However, constraints ensured that property keys were strictly mapped to IDS- and model-derived vocabulary, maintaining $SCR = 1.00$ across both models and avoiding confusion between similarly named attributes.
- **Level 3: Topology and relationships (hard).** Multi-hop traversals remained challenging. While the 7B model under S4 constraints achieved only marginal correctness ($EA \approx 0.03$), the 3B model under S4 provided the only successful executions in this category ($EA \approx 0.13$). This suggests that by narrowing the search space to valid relationship types and property keys, grammar constraints can occasionally assist smaller models in constructing complex graph patterns that would otherwise fail under unconstrained prompting.

Overall, grammar constraints substantially improve reliability for class and property retrieval by mitigating schema hallucinations, but they cannot compensate for limited logical reasoning in complex topological queries.

Limitations and implications

The results indicate that constrained NL2Cypher generation can substantially improve the reliability of natural-language querying over IFC-derived graphs, particularly for class retrieval and property-level queries. In practice, this can make BIM information more accessible to non-expert stakeholders by reducing the need to learn specialist query languages and by limiting common failure modes such as schema hallucinations and non-executable queries.

Several limitations must be acknowledged. First, the evaluation constitutes an initial validation on a single sample project with one IDS file and 16 queries; broader replication across IFC files from real projects, with different modelling conventions, varying data quality, and larger query sets, is required before generalisability claims can be made. Second, the hybrid schema is only as effective as the underlying IDS and the extracted model vocabulary; incomplete or overly permissive specifications may still allow ambiguous queries, while overly strict specifications can reduce coverage of valid (but unspecified) information. Third, and most critically, the framework exhibits near-total failure on Level 3 topology queries ($EA \approx 0.03$–0.13), demonstrating that grammar constraints—while effective at eliminating schema-level errors—cannot compensate for the multi-step logical reasoning that complex graph traversals require. Time and cognitive effort were not measured in this study; future work should include user studies to quantify usability gains.

Fourth, IDS availability is still uneven in current practice. The proposed framework is most effective in projects where at least a partial IDS already exists. When the IDS is incomplete, the hybrid schema can still enforce the entities and properties that are specified and rely on model-derived vocabulary elsewhere. When the IDS is missing entirely, the method loses its project-specific grounding and should be interpreted as a query-assistance setup rather than a full compliance-oriented one; grammar-constrained generation remains possible over the model-only vocabulary, but without the strict property typing and enumeration controls that prevent hallucinations on ambiguous or inconsistent data. The approach therefore does not replace careful IDS authoring, but it can benefit from gradual IDS formalisation: because the grammar makes explicit which entities and properties are queryable, project teams receive immediate feedback on specification gaps, turning IDS creation from a retrospective compliance exercise into a proactive step that directly improves query reliability.

Conclusions

This paper presented an IDS-constrained text-to-graph querying framework that addresses the reliability gap when applying LLMs to BIM information retrieval. By transforming IFC data into a labeled property graph and compiling project-specific IDS requirements into a hybrid schema, the approach enables grammar-constrained decoding that eliminates schema hallucinations and guarantees syntactically valid Cypher queries. The experimental results confirm that LLMs are most effective when used to generate query logic rather than retrieve information directly, and that combining graph-based representations with formal semantic constraints improves accuracy.

These findings have practical implications for the Architecture, Engineering, Construction and Operators (AECO) industry. First, LLM-mediated BIM querying should be treated as a query generation problem with hard constraints, not a prompt engineering problem. Second, IDS can serve proactively to reduce variability in query formulation, beyond its traditional role in retrospective compliance checking. Third, by offloading syntactic precision to the constraint engine, smaller open-weights models can achieve reliability levels previously reserved for large cloud models, enabling privacy-preserving, local deployment without uploading sensitive project data to external APIs.

Future work will (i) scale experiments across additional domains and IDS complexities, (ii) investigate robustness under non-compliant or inconsistent IFC models, and (iii) extend the graph schema to better represent spatial context (e.g., second-level space boundaries) for queries about connectivity, adjacency, and energy-relevant relationships. Beyond accuracy, future evaluations should consider deployment factors such as latency and the environmental footprint. Two promising directions are interactive self-correction, where execution feedback (e.g., "query returned 0 nodes") drives iterative refinement, and hybrid graph-vector retrieval for unstructured data. Nevertheless, the results demonstrate that this NL2Cypher technique provides a practical and effective pathway toward democratizing BIM information access in a reliable way.

References

Autodesk (2026). Autodesk (company website). `https://revit.downloads.autodesk.com/download/2024RVT_RTM/Docs/InProd/racadvancedsampleproject.rvt`. Accessed: 2026-03-23.

Beetz, J., van Leeuwen, J. P., and de Vries, B. (2009). IfcOWL: A case of transforming EXPRESS schemas into OWL ontologies. In Proceedings of the 1st International Workshop on Semantic Digital Product Memories (SDPM 2009). Co-located with the 6th European Semantic Web Conference (ESWC 2009).

Bloch, T., Borrmann, A., and Pauwels, P. (2023). Graph-based learning for automated code checking–exploring the application of graph neural networks for design review. Advanced Engineering Informatics, 58:102137.

Borrmann, A., Beetz, J., Koch, C., Liebich, T., and Muhic, S. (2018). Industry foundation classes: A standardized data model for the vendor-neutral exchange of digital building models. In Borrmann, A., König, M., Koch, C.,

and Beetz, J., editors, Building Information Modeling, chapter 5. Springer, Cham.

buildingSMART International (2024). Information Delivery Specification (IDS). Standard and documentation. Accessed 2026-01-30.

Daum, S. and Borrmann, A. (2014). Processing of topological BIM queries using boundary representation based methods. Advanced Engineering Informatics, 28(4):272–286.

dottxt-ai (2026). Outlines documentation. `https://dottxt-ai.github.io/outlines/latest/`. Accessed 2026-02-05.

Du, C., Esser, S., Nousias, S., and Borrmann, A. (2026). Text2bim: Generating building models using a large language model-based multiagent framework. Journal of Computing in Civil Engineering, 40(2):04025142.

Esser, S., Vilgertshofer, S., and Borrmann, A. (2022). Graph-based version control for asynchronous BIM collaboration. Advanced Engineering Informatics, 53:101664.

Gao, Y., Xiong, Y., Gao, X., Jia, K., Pan, J., Bi, Y., Dai, Y., Sun, J., Wang, H., and Wang, H. (2023). Retrieval-augmented generation for large language models: A survey. arXiv preprint arXiv:2312.10997, 2(1).

Hellin, S., Nousias, S., and Borrmann, A. (2025). Natural language information retrieval from BIM models: An LLM-based agentic workflow approach.

Hjelseth, E. and Nisbet, N. (2016). Model checking in BIM-based projects. International Journal of 3-D Information Modeling, 5(2):1–20.

Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Liu, X., Wang, W., and Feng, H. (2023). A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. arXiv preprint arXiv:2311.05232.

International Organization for Standardization (2024). ISO 16739-1:2024 — industry foundation classes (IFC) for data sharing in the construction and facility management industries — part 1: Data schema. `https://www.iso.org/standard/84123.html`. Accessed 2026-02-05.

Iranmanesh, S., Saadany, H., and Vakaj, E. (2025). LLM-assisted Graph-RAG information extraction from IFC data. arXiv preprint arXiv:2504.16813.

Ismail, A., Nahar, A., and Scherer, R. (2017). Application of graph databases and graph theory concepts for advanced analysing of BIM models based on IFC standard. Proceedings of EGICE, 161.

Jiang, H. et al. (2023). StructGPT: A general framework for large language model to reason over structured data. arXiv preprint arXiv:2305.09645.

Khademi, H. and Behan, A. (2017). A review of approaches to solving the problem of BIM search: towards intelligence-assisted design. In Proceedings of CitA BIM Gathering 2017, Dublin, Ireland.

Mazairac, W. and Beetz, J. (2013). BIMQL–an open query language for building information models. Advanced Engineering Informatics, 27(4):444–456.

Neo4j (2026). Neo4j graph database platform. `https://neo4j.com/`. Accessed 2026-02-05.

Pauwels, P., Krijnen, T., Terkaj, W., and Beetz, J. (2017). Enhancing the ifcOWL ontology with an alternative representation for geometric data. Automation in Construction, 80:77–94.

Rasmussen, M. H., Hviid, C. A., and Karlshøj, J. (2020). Bot: The building topology ontology of the w3c linked building data group. Semantic Web, 11(4):535–559.

Solihin, W. and Eastman, C. (2015). Classification of rules for automated BIM rule checking development. Automation in Construction, 53:69–82.

Wang, Z., Sacks, R., and Yeung, T. (2022). Exploring graph neural networks for semantic enrichment: Room type classification. Automation in Construction, 134:104039.

Yin, M., Tang, L., Webster, C., Xu, S., Li, X., and Ying, H. (2023a). An ontology-aided, natural language-based approach for multi-constraint BIM model querying. Journal of Building Engineering, 76:107066.

Yin, Y., Heaton, J., Borrmann, A., and Beetz, J. (2023b). An ontology-based approach for natural language querying of building information models. arXiv preprint arXiv:2303.15116.

Yu, Y. et al. (2023). Trends and advances in IFC schema extensions for BIM interoperability: A systematic review. Applied Sciences, 13(23):12560.

Zhang, C., Beetz, J., and de Vries, B. (2018). BimSPARQL: Domain-specific functional SPARQL extensions for querying RDF building data. Semantic Web, 9(6):829–855.

Zhu, J., Nisbet, N., Yin, M., Wei, R., and Brilakis, I. (2025). Releasing the power of graph for building information discovery. Automation in Construction, 172:106034.

Zhu, J., Wu, P., and Lei, X. (2023). IFC-graph for facilitating building information access and query. Automation in Construction, 148:104778.