Automation in Construction 188 (2026) 107003

Contents lists available at ScienceDirect

Automation in Construction

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

LLM-enabled multi-agent framework for automated Scan-to-BIM and Scan-to-Graph reconstruction ☆

Yuandong Pan [a], Mudan Wang [b],*, Jiechao Gao [a], Jie Wang [a], Michael D. Lepech [a], Ioannis Brilakis [b]

[a] School of Engineering, Stanford University, Y2E2 Building, 473 Via Ortega, Stanford, 94305, CA, United States
[b] Department of Engineering, University of Cambridge, 7a JJ Thomson Avenue, Cambridge, Cambridgeshire, United Kingdom

ARTICLE INFO

Keywords:
Building Information Modeling (BIM)
Large Language Models (LLMs)
Scan-to-BIM
Scan-to-Graph
Digital twin
Multi-agent framework

ABSTRACT

BIM and semantic graphs are increasingly used in the AEC industry. Converting indoor point clouds into standards-compliant IFC BIM and graph representations remains non-trivial and often requires expert knowledge and manual post-processing steps. To address this, an LLM-enabled multi-agent framework is proposed to automate Scan-to-BIM and Scan-to-Graph from point clouds, with support for natural-language, intent-driven reconstruction. The framework integrates query interpretation, point-cloud inference, topology-aware IFC generation, semantic graph construction, and multi-stage validation into a single end-to-end pipeline, explicitly handling of wall–wall junctions and wall–slab alignment. The framework is evaluated on three datasets captured with different sensing modalities (mobile laser scanning, RGB-D, and synthetic). On the TUMCMS test set, a laser-scanned point cloud dataset captured in office environments at the Technical University of Munich, room areas of generated IFC files achieve mean absolute deviations of $1.45m^2$ (Llama-based) and $1.51m^2$ (Qwen-based), corresponding to mean relative deviations of 4.70% and 5.10% versus manually created BIM. The results demonstrate robust and generalized performance of the framework for wall and space reconstruction across datasets.

1. Introduction

The architecture, engineering, and construction (AEC) industry is increasingly using information-rich digital representations of physical assets in various use cases. Building Information Modeling (BIM) and, more recently, digital twins provide structured geometric and semantic descriptions of buildings that extend a wide range of use cases, including space management, asset management, compliance checking, maintenance scheduling, energy simulation and analysis, and lifecycle assessment [1]. If updated regularly, these models can create substantial value for all stakeholders by improving decision-making in almost all use cases.

In practice, a large proportion of the built environment still lacks accurate and up-to-date digital representations. Many buildings were constructed before BIM was proposed. Even when BIM models exist for some facilities, they are frequently out of date due to activities like renovations and component replacement that are not reflected in the digital models. This problem limits the applicability of BIM and digital twin workflows, particularly for existing assets where retrofit and operational work plans are most needed.

Recent advances in sensing technologies provide an opportunity to solve this problem partially. Data collection devices, such as mobile laser scanners and RGB-D cameras, can capture the environments rapidly and produce dense point clouds that include accurate geometric information. In parallel, point cloud deep learning has also become more and more mature, allowing semantic segmentation, instance detection, and geometric primitive extraction for common indoor elements [2]. These advancements make it increasingly feasible to reconstruct building components and their relationships directly from captured data [3].

Despite the progress in sensing technologies and point cloud deep learning, converting semantically enriched point clouds into a standardized data format for buildings remains non-trivial. While a labeled point cloud is not a BIM model, IFC-based BIM requires explicit parametric entities, correct spatial and topological information (like wall connectivity, window/door openings embedded in walls, and slabs aligned to wall faces). It introduces extra complexity to make sure that the reconstructed geometry is compliant with schema constraints.

☆ This article is part of a Special issue entitled: 'Next-Gen Construction' published in Automation in Construction.

* Corresponding author.
E-mail address: mw923@cam.ac.uk (M. Wang).

https://doi.org/10.1016/j.autcon.2026.107003
Received 15 December 2025; Received in revised form 27 April 2026; Accepted 29 April 2026
Available online 7 May 2026

In addition to BIM, graph-based representations are also attracting attention in the AEC industry as they naturally support heterogeneous entities (spaces, elements, systems), explicit relationships (adjacency, containment, connectivity), and flexible querying. Semantic graphs can complement IFC by demonstrating relationships explicitly that are useful for applications such as facility management. However, constructing a high-quality semantic graph from point cloud data is also challenging [4]. It requires processing point cloud data properly, making consistent entity definitions, extracting relationships robustly, and validating to avoid propagating potential errors into the graph database.

Large Language Models (LLMs) provide an opportunity to introduce higher-level interaction mechanisms into reconstruction workflows. By enabling natural-language interaction, LLMs allow users to specify reconstruction intent (e.g., desired level of detail or target elements), without directly interacting with low-level parameters or BIM schema definitions. This shifts the reconstruction process from parameter-driven configuration to intent-driven specification, providing a structured way to control reconstruction outputs.

This paper focuses specifically on the automatic reconstruction of indoor architectural environments from indoor point clouds. The target elements are primarily room-scale architectural components, including walls, floors, ceilings, openings, spaces, and optionally furniture. The study does not address exterior reconstruction, structural analysis, or Mechanical, Electrical, and Plumbing (MEP) system reconstruction. By defining this scope explicitly, the paper concentrates on the creation of usable indoor digital representations for graph and BIM-related applications.

This study addresses the following research questions: (a) How can user intent described in natural language be integrated into an automated indoor Scan-to-BIM workflow so that non-expert users can control reconstruction outputs without manual parameter setting? (b) How can point clouds of indoor environments be converted automatically into IFC models while preserving essential topological relationships such as wall–wall junctions and wall–slab alignment? (c) How can the same reconstructed scene be transformed into a semantic graph representation that complements IFC by making spatial and semantic relationships explicit and queryable? (d) To what extent does the proposed framework generalize across indoor point clouds captured with different sensing modalities in a zero-shot setting?

Accordingly, the objectives of this paper are formalized as follows: (a) to develop an LLM-enabled multi-agent framework for indoor Scan-to-BIM and Scan-to-Graph reconstruction; (b) to enable intent-driven reconstruction through natural-language queries; (c) to generate standards-compliant IFC models with a designed approach of topological constraints; (d) to construct a semantic graph from the same reconstructed scene; and (e) to evaluate the framework across datasets acquired with different sensors.

To achieve the objectives of converting point cloud data to BIM and graph-based representations and utilizing the benefits of natural-language interaction provided by LLMs, we propose an LLM-enabled multi-agent framework that automates the process of creating IFC BIM models and a corresponding semantic graph from point cloud data.

In this study, Scan-to-BIM and Scan-to-Graph are treated not as two independent pipelines, but as two complementary representations derived from the same reconstructed indoor scene: IFC provides standards-compliant geometric and semantic building information, while the graph provides an explicit relational abstraction for querying, reasoning, and other analytics. The main contributions of this paper are listed as follows:

- An intent-driven reconstruction paradigm that maps natural-language user queries into structured reconstruction configurations, enabling flexible control over the reconstruction scope and level of detail.
- A novel topology-aware IFC generation method that handles wall–wall junctions and wall–slab alignment.
- A unified pipeline that generates both IFC models and semantic graphs from a shared intermediate representation, presenting geometric and relational information of indoor environments.
- A hybrid multi-agent framework that integrates LLM-based reasoning with deterministic geometric and validation modules, improving adaptability and reliability of the reconstruction process.

The rest of the paper is organized as follows. Section 2 reviews related work on scan-to-BIM reconstruction, graph-based building representations, and the adoption of LLMs in the AEC domain. Section 3 details the proposed multi-agent framework, including the design of various agents and the end-to-end workflow. Section 4 presents and discusses experimental results on three datasets with different sensing modalities. Finally, Section 5 concludes with a summary of the main contributions, limitations, and practical implications.

2. Research background

This section organizes the literature review into three areas: Scan-to-BIM, graph representation of buildings, and LLM in the AEC industry.

2.1. Scan-to-BIM

Scan-to-BIM research has primarily focused on reconstructing core indoor building components, including walls, floors, ceilings, and openings, because these elements form the backbone of BIM representations and enable downstream modeling and analysis. At a broader level, scan-to-BIM has been formalized as an application workflow that outlines the key steps from data capture to BIM generation and specifies accuracy requirements for common building elements [5]. Within this context, a common approach is to segment point clouds into meaningful regions and assign semantic labels to key structural surfaces, even in cluttered scenes [6,7]. Building on this, some methods first detect planar geometries and then classify them by combining geometric features with contextual cues, which helps recognize visible components such as walls, ceilings, windows, and doors [8,9]. To improve robustness in real projects, other studies reduce the need for scanner trajectory information or strong acquisition priors, and instead reconstruct rooms and openings directly from point clouds under weaker sensing assumptions [10].

Advances in 3D deep learning have further shifted the focus from rule-based segmentation towards data-driven prediction of building elements from point clouds. In parallel, indoor layout reconstruction has attracted attention as BIM generation depends on accurate spatial interpretation. Recent work includes converting point clouds into 2D floor plan representations to describe room layout and wall arrangements [11], restoring floor plan topology by extracting wall points, fitting line segments via iterative RANSAC-style clustering, and assembling them into vectorized layouts [12], starting with reconstructing room spaces [13], and reconstructing 3D wireframe models by extracting and aligning boundary primitives under weak Manhattan assumptions [14]. Deep learning pipelines have also been proposed to recover multi-room environments while modeling walls and volumetric openings, producing semantically segmented representations that can support BIM creation [15]. More integrated scan-to-BIM pipelines further combine semantic parsing with parametric modeling procedures to support diverse room configurations and to output BIM elements in authoring tools [16]. Construction scenarios have also been investigated, where laser-scanned point clouds are used to generate as-built BIMs for partially completed structures by segmenting elements such as slabs, beams, columns, and walls without requiring an as-planned reference model [17].

Compared with structural and layout reconstruction, scan-to-BIM for non-structural elements remains less developed, despite its importance for coordination and operations. Existing studies address MEP reconstruction through pipelines that detect components [18], extract geometric parameters, infer connection relationships [19,20], and generate parametric BIM representations [21], as well as through geometry-driven segmentation strategies such as region growing [22]. However, modeling is often more reliable for simple regular components than for heterogeneous fittings and terminals, and manual intervention is still frequently required to achieve complete systems [23]. Beyond MEP systems, indoor objects such as furniture have been reconstructed using both template-based and learning-based methods [24,25]. In addition, synthetic point clouds are used to improve the performance of learning-based methods in recognizing these object categories [26,27]. Apart from point cloud processing, a persistent challenge is schema mapping and export, since transferring segmented point clusters into semantically complete IFC representations remains difficult, particularly for less common MEP entities, which may lead to incomplete model instantiation and information loss [17].

Overall, prior studies demonstrate substantial progress in element recognition and layout reconstruction from point clouds, yet limitations remain in highly occluded environments and under domain shift. More fundamentally, much of the literature emphasizes element-level identification, while the explicit modeling of spatial and functional interrelationships among components, such as containment, adjacency, connectivity, and hosting, has received comparatively limited attention. These interrelationships are essential for generating structured and queryable BIM models that support building management tasks [28].

2.2. Graph representation of buildings

Graph-based data models are increasingly adopted to represent buildings and their digital twins because they enable modular modeling, extension, and the encoding of rich relationships, including adjacency, containment, and process dependency [29,30]. By treating relationships as essential information, graph representations help integrate heterogeneous building data that would otherwise remain fragmented across BIM models, sensor systems, schedules, and domain databases. Across construction and operation, graphs often work as an integration backbone and a foundation for analytics. For example, some previous studies have focused on the process of Scan-to-Graph of indoor environments, integrating semantic information extracted from images [31].

Knowledge graphs have been coupled with perception pipelines. For example, graph-enhanced computer vision has been used to improve the robustness and transferability of concrete pouring monitoring, illustrating how relational context can strengthen process digital twinning [32]. Graph representations also support learning-based semantic enrichment. Room classification on apartment layouts shows that transforming layouts into graphs enables graph neural networks for topology-aware inference [33]. Graph-based data exchange has also been explored to support operational workflows, including transferring information from BIM repositories to robotic systems for construction and facility tasks [34]. For instance, some studies link BIM models with IoT metadata and process information through a knowledge graph to support consistent data ingestion and unified access in digital twins [35].

IFC-derived graph representations have received particular attention, partly because they offer a structured route for converting BIM data into queryable relational forms. Zhu et al. analyze the internal structure of IFC graphs and propose a five-level level-of-detail framework for generating graphs from IFC. Their findings suggest that IFC relations form the backbone for efficient relational querying and that controllable graph granularity can improve both query performance and knowledge discovery [3,36]. This is highly relevant for building digital twins because graph construction choices influence the effectiveness of information retrieval and automated analysis. IFC-based property graphs have also been used to support building performance workflows. Iliadis et al. present a workflow that connects BIM data to building energy modeling via an IFC graph representation and Modelica-based simulation. They argue that explicit relational structure improves query efficiency, supports clearer inspection of connected data, and helps reveal inconsistencies in BIM information [37].

Graph representations are also used in building reconstruction and scene understanding, where relational structure is extracted from geometry data. Some approaches detect rooms and spatial relations through intermediate graph abstractions, while geometric deep learning methods leverage topology graphs to learn relationships between labeled building surfaces [38,39]. Nevertheless, much of the graph-based reconstruction literature has focused on city-scale or outdoor environments, typically using airborne laser scanning and urban datasets [40]. In comparison, relatively few studies address the generation of fine-grained indoor semantic graphs from indoor point cloud data. This limitation is significant because indoor digital twins and facility management applications depend on explicit room-scale relationships.

2.3. LLM in AEC

LLMs are increasingly adopted across architecture, engineering, and construction as natural-language interaction layers for accessing multimodal project information. Their ability to interpret text, images, and structured representations is enabling new workflows for design support, BIM interaction, site monitoring, compliance checking, and digital twin operations.

LLM-enabled BIM has been explored as a means of translating conversational intent into modeling actions and detailing decisions, thereby reducing reliance on conventional interaction procedures. Jang et al. illustrate an LLM–BIM chaining approach for architectural detailing that separates layer specification from creation and applies structured prompting to improve engineering rationality, with validation indicating that the generated outputs can satisfy user requirements while remaining broadly consistent with engineering expectations [41]. Beyond detailing, LLMs have been positioned to support prefabrication-oriented practices by strengthening knowledge use and decision making in Design for Manufacture and Assembly [42]. At a broader scale, LLM-enabled question answering has been used to capture requirements and constraints, which are then resolved across disciplines through a prioritized optimization mechanism [43].

Natural-language access to BIM data has also been developed to support spatial reasoning and reduce the need for explicit knowledge of BIM schemas and query languages. Early assistant implementations focus on prompt-based interpretation of user queries to retrieve and present information from BIM repositories [44]. Subsequent work exports BIM data into databases and translates natural-language questions into executable queries, using vector retrieval to supply relevant schema and context under token limitations [45]. Building on these foundations, multi-agent systems have been proposed to decompose requests into specialized roles, such as aligning intent with BIM functions and routing tasks for retrieval or code generation [46]. For spatial reasoning, designed systems extract geometric information from openBIM models, construct spatial indices, and coordinate agent collaboration to answer complex spatial relationship queries [47]. While these methods improve usability, the performance relies on grounding in explicit geometry and metadata and on verifiable tool execution.

LLMs are also being applied to compliance and regulation workflows. Retrieval-augmented generation has been used to support conversational quality checks against regulatory texts, where hybrid retrieval improves evidence matching and justification [48]. Complementary work combines deep learning classification of regulatory content with LLM-based extraction to reduce annotation requirements and improve structured information capture from codes [49]. In parallel, LLM-driven compliance checking has been integrated with BIM authoring environments, enabling models to interpret code requirements, generate checking scripts, and support semi-automated checking and reporting [50]. These studies indicate efficiency gains, but also underscore the need for rule handling and auditable evidence trails.

In digital twin and operations contexts, LLMs are often treated as an intelligence layer for interaction, integration, and decision support [51]. Agent-based intelligent digital twins formalize data and services through a digital twin ontology to enable agent reasoning and tool-based task execution during operations [52]. From an information management perspective, GPT-based assistants have been demonstrated for helping stakeholders define operation and maintenance information requirements and generate guidelines aligned with those requirements [53]. At a broader systems level, frameworks and surveys discuss LLM-enabled digital twins for human-in-the-loop cyber–physical and IoT systems and propose taxonomies linking digital twin functions to LLM capabilities such as description, prediction, and prescription [54,55]. In parallel, digital twin development frameworks highlight unresolved deployment challenges, including scalability, privacy, and hallucination control [56]. Finally, conversational interfaces have been explored to support worker–robot collaboration through integration of voice commands, robotics stacks, and BIM-based context [57].

Although LLMs have shown strong potential in BIM interaction, compliance checking, and digital twin reasoning, their role in point-cloud-based reconstruction should be framed carefully. In the context of this paper, the LLM is not used as a standalone geometric reconstruction engine operating directly on raw point clouds. Instead, point cloud understanding is performed by a specialized foundation model (SpatialLM [58]), while the LLM contributes at the system level by interpreting user intent, configuring reconstruction scope, coordinating agents, and structuring outputs in a form suitable for IFC and graph generation. This is important because it positions LLMs as an interaction and orchestration layer for automation, rather than as a replacement for geometric inference models.

To summarize, existing Scan-to-BIM pipelines face three key limitations. First, most approaches focus on geometric reconstruction but lack a mechanism to explicitly control reconstruction scope and level of detail in a flexible and interpretable manner. Second, while IFC generation methods exist, they often fail to ensure topological consistency (e.g., wall connectivity and alignment), requiring manual correction. Third, although graph representations have been explored, there is no unified framework that derives both BIM and relational graph representations consistently from the same reconstructed scene.

More fundamentally, current pipelines lack an explicit mechanism to map high-level user intent into structured reconstruction configurations. This limits the adaptability of reconstruction systems, as users must rely on predefined parameters or manual post-processing rather than simply prompting specifications of desired outputs.

Recent studies have explored multi-agent frameworks to decompose complex AEC tasks into specialized components, enabling modular reasoning and task orchestration. These systems have been applied to BIM querying, compliance checking, and digital twin operations, where agents collaborate to interpret user intent, retrieve information, and execute domain-specific functions. Such approaches demonstrate improved flexibility and scalability compared to monolithic pipelines.

However, several limitations remain. First, many existing multi-agent frameworks focus primarily on information retrieval or decision support, rather than tasks such as Scan-to-BIM reconstruction, where strict geometric consistency and schema compliance are required. Second, LLM-based agents are frequently used without a clear separation between reasoning and deterministic processing, which may reduce reliability in geometry-sensitive tasks.

To address these limitations, this paper proposes an intent-driven reconstruction framework that integrates LLM-based query interpretation with deterministic IFC generation and semantic graph construction.

3. Proposed approach

This section presents the proposed LLM-enabled multi-agent framework for Scan-to-BIM and Scan-to-Graph reconstruction. It first provides an overview of the overall system architecture, followed by detailed descriptions of each agent.

Table 1
Implementation of agents in the proposed framework.

Agent	Key mechanism
Query interpreter	Structured JSON prompt to local LLM
Inference	Fixed template prompt to SpatialLM (not conversational)
IFC configuration	Deterministic algorithm
Graph generation	String parsing + graph construction
Validation	Rule-based checks

3.1. Framework overview

This paper proposes a multi-agent framework that achieves fully automatic generation of point cloud data to IFC representations and semantic scene graphs through interaction in natural language. Unlike other reconstruction pipelines from point cloud data, our framework uses LLMs to interpret user queries, analyze user intent, and orchestrate specialized agents for autonomous reconstruction.

The framework is illustrated in Fig. 1. There are five agents to process input data (point cloud data and user query), generate room layouts, and create BIM models in IFC format together with semantic graphs in JSON format. These includes: (1) Query Interpreter Agent that translates user queries in natural language into structured reconstruction configuration using LLM's capability of reasoning; (2) Inference Agent that achieves point cloud processing and structural element detection; (3) IFC Configuration Agent that generates IFC files with configurable geometric and semantic specifications from user input; (4) Graph Generation Agent that creates scene graphs representing spatial relationships and hierarchical relationships for reconstructed models; and (5) Validation Agent performs quality assurance checks at three critical pipeline stages to ensure output reliability, including validating input point cloud integrity, verifying generated layout completeness, and confirming IFC schema compliance. In addition, there is a Coordinate Agent that orchestrates the entire pipeline, managing inter-agent connections, to deliver reconstruction results.

It should be noted that the agents in the proposed framework are implemented using a hybrid design rather than a uniform architecture. Specifically, the Query Interpreter Agent is implemented as an LLM-based module using structured prompting to map natural-language queries into machine-actionable configurations. In contrast, the Inference Agent relies on a vision-language foundation model (SpatialLM) with a fixed template prompt for point cloud understanding, while the IFC Configuration Agent, Graph Generation Agent, and Validation Agent are implemented as deterministic, rule-based procedures. This design, shown in Table 1, reflects the different roles of agents in the pipeline: LLM-based agents are used for intent understanding and flexible reasoning, while algorithmic agents are used for geometry processing, schema-compliant modeling, and validation to ensure reliability and reproducibility.

3.2. Coordinator agent

The Coordinator Agent works as the orchestration layer that turns individual agents into a single, reliable reconstruction workflow. It manages inter-agent connections, handles validation checkpoints, and propagates error information.

Generally, the coordinator runs a seven-stage pipeline with validation points between major transformation steps. Stage 1 performs input validation and point-cloud quality assessment to confirm that the data are readable and suitable for reconstruction. Stage 2 interprets the user's natural-language query, converting it into structured reconstruction specifications. Stage 3 executes model inference, generating the layout code from the point cloud using the SpatialLM foundation model. Stage 4 performs layout validation to ensure the generated layout is geometrically valid and complete. Stage 5 exports the validated layout into an IFC-based BIM model at the requested reconstruction

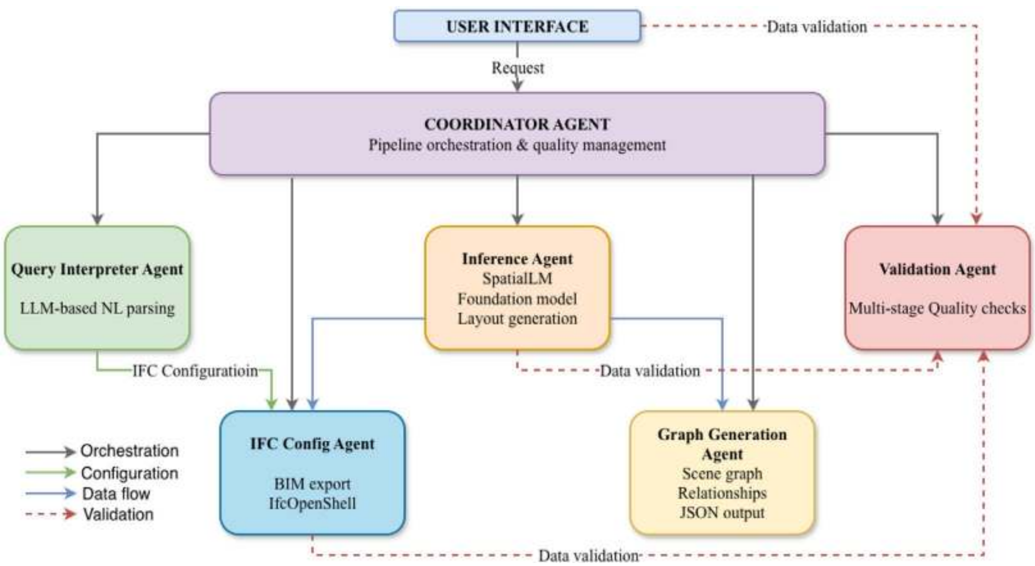


Fig. 1. System architecture of the LLM-enabled multi-agent framework for Scan-to-BIM and Scan-to-Graph reconstruction.

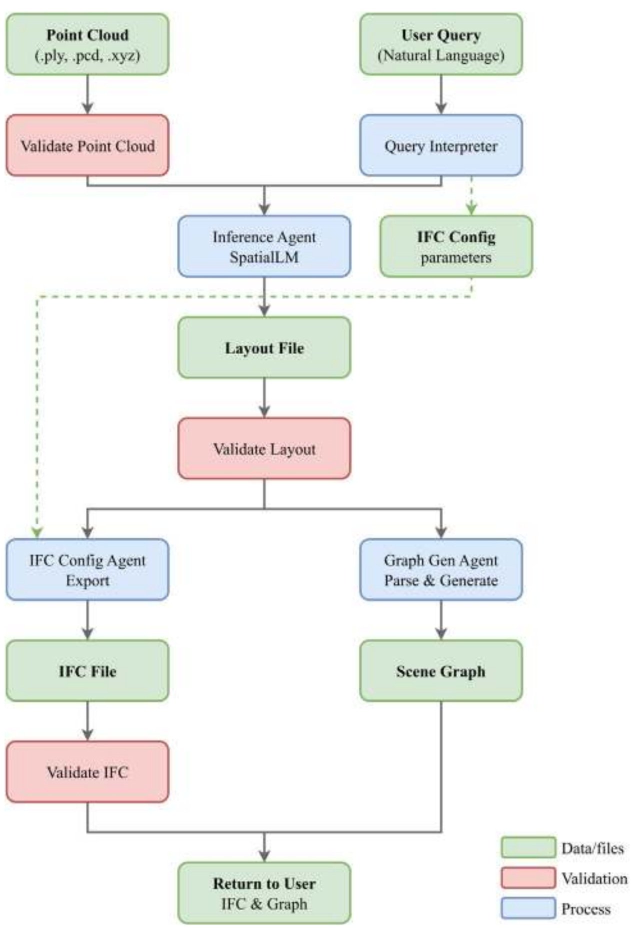


Fig. 2. Data flow diagram of the multi-agent framework.

level. Stage 6 checks the generated IFC file for schema compliance. Stage 7 generates the scene graph representation and extracts the corresponding semantic relationships. The dataflow orchestrated by the agent is illustrated in Fig. 2, while Table 2 summarizes the key inputs and outputs of each agent.

Error handling is also considered at these stages. Hard failures (like unsupported format of input files, inference exceptions, or IFC schema violations) terminate the pipeline and return detailed messages describing the failure mode. Soft failures, such as graph generation errors, allow the pipeline to proceed to generate an IFC file without the graph output. In addition, warnings are recorded and reported to users without interrupting execution, improving transparency while preserving usability.

3.3. Query interpreter agent

The Query Interpreter Agent achieves an important feature in reconstruction fields like Scan-to-BIM technology that allows non-expert users to specify reconstruction requirements through natural language. By employing LLMs to analyze user input, the agent automatically translates user queries such as "reconstruct the complete room without furniture" or "I need just the walls and floor" into structured, machine-actionable reconstruction configurations.

The agent implements five reconstruction levels that correspond to different representations of indoor environments. The purpose of defining these levels is to support different use cases and allow non-expert users to control reconstruction complexity through natural-language queries. In many practical scenarios, users do not require a full architectural BIM model. For example, facility managers may only need wall layouts for spatial analysis, while renovation planning may require doors and windows but not furniture. The five levels represent the level of detail in the generated IFC model. The *walls_only* level extracts only wall structures without openings, floors, or ceilings, which is suitable for basic spatial layout analysis. The *walls_floor* level adds horizontal floor slabs while maintaining simplified walls, enabling clearer visualization of room extents without occlusion from ceilings. The *basic_room* level produces a complete structural room reconstruction including walls, floors, ceilings, and spaces, where openings are represented as voids in wall geometry using `IfcOpeningElement`. The *full_arch* level extends this representation by generating parametric door and window objects (`IfcDoor`, `IfcWindow`). Finally, the *with_furniture* level adds detected furniture as `IfcFurnishingElement` entities, producing the most complete reconstruction.

The Query Interpreter Agent maps natural-language user queries to one of these reconstruction levels. For example, the user query "reconstruct the room layout with walls only" will be interpreted as *walls_only*, while "generate a full BIM model with doors and windows"

Table 2
Inputs and outputs of agents in the proposed framework.

Agent	Input	Output
Query interpreter	Natural-language user query	Structured JSON configuration (reconstruction level, detection type, etc.)
Inference	Point cloud + reconstruction configuration	Layout code (parametric primitives: walls, doors, etc.)
IFC configuration	Layout code + reconstruction configuration	IFC BIM model (IfcWall, IfcDoor, IfcSlab, etc.)
Graph generation	Layout code	Semantic scene graph (in JSON format)
Validation agent	Point cloud/layout/IFC model (at different stages)	Validation reports (errors, warnings, consistency checks)

corresponds to *full_arch*. The resulting structured configuration is then passed to further steps to generate IFC files.

The Query Interpreter Agent is implemented entirely through LLM prompting. The full prompt template that clarifies the reconstruction levels and constrains the model to respond with valid JSON aligned to a predefined schema is provided below:

System Prompt

You are an expert AI assistant for 3D building reconstruction from point cloud data.
Your task is to analyze user queries and determine what building elements should be reconstructed.
Available reconstruction levels:
1. walls_only - Only wall structures, no openings, floors, or ceilings
2. walls_floor - Walls and floor slabs, no ceilings or openings
3. basic_room - Walls, floor, ceiling, and spaces, with openings (voids) but no door/window objects
4. full_arch - Complete architecture: walls, floor, ceiling, door objects, window objects, and spaces
5. with_furniture - Everything including all furniture and fixtures
You must respond ONLY with a valid JSON object in this exact format:
{
"reconstruction_level": "one of: walls_only, walls_floor, basic_room, full_arch, with_furniture",
"reasoning": "brief explanation of your decision",
"confidence": 0.9,
"categories": ["list of specific furniture categories if mentioned"]
}
DO NOT include any text outside the JSON object.

The Query Interpreter Agent also produces a confidence score associated with its output. In the current implementation, this score reflects the model's self-reported certainty in generating the structured configuration. However, this confidence value is not used to control downstream processing or decision-making in the pipeline. Instead, it is provided as auxiliary information that may support future extensions, such as human-in-the-loop workflows or uncertainty-aware processing. A rigorous definition and evaluation of confidence estimation in LLM-based reconstruction remains an open research problem and is left for future work.

3.4. Inference agent

The Inference Agent processes the input point cloud and estimates the geometric parameters required for downstream IFC generation and semantic scene-graph construction. Given an input point cloud $P = \{(\mathbf{p}_i, \mathbf{c}_i)\}_{i=1}^{N}$ with coordinates $\mathbf{p}_i \in \mathbb{R}^3$ (and color $\mathbf{c}_i \in \mathbb{R}^3$), the agent infers a room layout expressed as a set of parametric elements in a predefined schema, including walls, openings, doors, windows, and furniture bounding boxes. In this work, this capability is implemented using the SpatialLM model, which integrates a 3D point cloud encoder with a language model to perform language-conditioned spatial reasoning and generate layout code.

The inference pipeline begins with multi-stage point cloud preprocessing to transform user-uploaded scan data into the representation expected by the SpatialLM model. The agent supports industry-standard point cloud formats, including PLY (Polygon File Format), PCD (Point Cloud Data), and ASCII XYZ. The preprocessing steps consist of several operations, including steps such as voxelizing point cloud data, normalizing RGB values, and translating point coordinates to non-negative values.

SpatialLM uses both discrete voxel indexing for efficient spatial aggregation and continuous metric coordinates for precise geometric prediction. It performs grid sampling with spatial hashing to map each point to a discrete grid coordinate $\mathbf{g}_i \in \mathbb{Z}^3$ while retaining the corresponding continuous coordinate $\mathbf{p}'_i \in \mathbb{R}^3$ after normalization. The final per-point feature vector concatenates grid coordinates, continuous coordinates, and color as $\mathbf{x}_i = [\mathbf{g}_i, \mathbf{p}'_i, \mathbf{c}_i] \in \mathbb{R}^9$, forming a unified representation $\mathbf{X} \in \mathbb{R}^{N\times 9}$. This hybrid encoding enables the model to exploit voxel-based spatial structure without losing metrics required for element dimensions and placements.

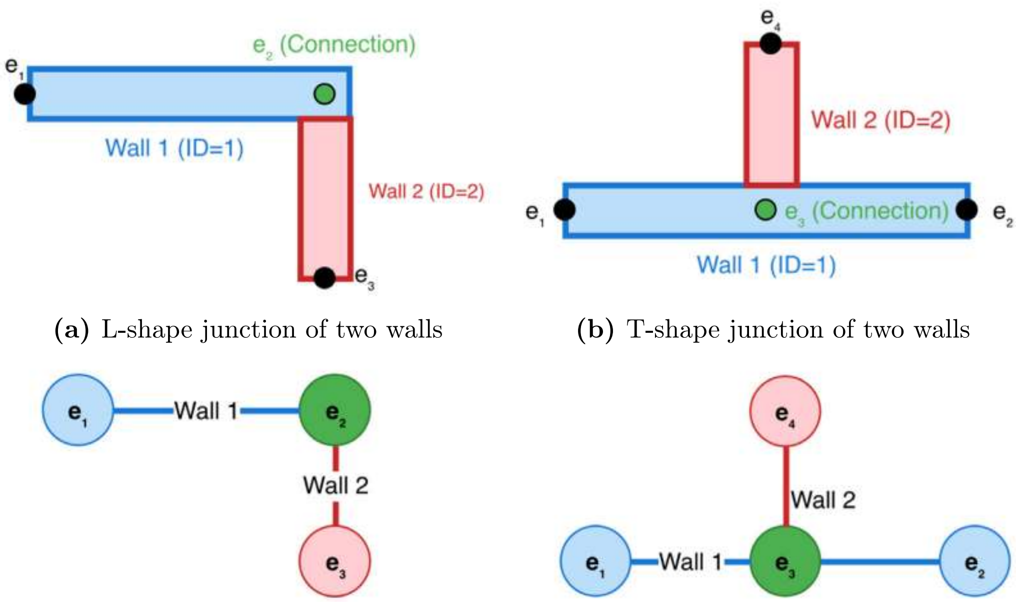
Rather than directly getting geometric primitives via traditional neural networks, SpatialLM produces scene layouts through constrained code generation. The model takes the encoded point features together with a task instruction prompt as input, and decodes a sequence of tokens that follow a fixed code grammar for parametric primitives. Walls are represented as `wall_k = Wall(ax, ay, az, bx, by, bz, height, thickness)`, while doors and windows are attached to host walls via `door_m = Door(wall_k, x, y, z, width, height)` and `window_n = Window(wall_k, x, y, z, width, height)`. When furniture is requested, objects are represented using oriented 3D bounding boxes as `bbox_t = Bbox(class_name, x, y, z, angle, sx, sy, sz)`. The agent parses the generated code into a structured intermediate representation that stores element identifiers, parameters, and relationships such as wall associations for openings, providing a deterministic and machine-actionable interface for both IFC generation and scene-graph construction.

Post layout generation, the agent applies inverse transformations to map generated coordinates back to the original point cloud, ensuring geometric consistency between the input point cloud and generated BIM elements in downstream IFC and graph generation.

3.5. IFC configuration agent

The IFC Configuration Agent converts LLM-generated layout representations to standard BIM formats. It implements the ISO 16739 Industry Foundation Classes (IFC) schema for digital building representation. This agent outputs IFC models compatible with commonly used AEC software platforms such as Revit, ArchiCAD, Tekla, and Solibri. The IfcOpenShell library, an open-source Python toolkit for IFC generation, is used to construct IFC files with full parametric geometry, spatial relationships, and property sets.

Unlike fixed-schema exporters that generate the same IFC files regardless of input, the agent dynamically adjusts IFC entity creation based on reconstruction level derived from user intent. The



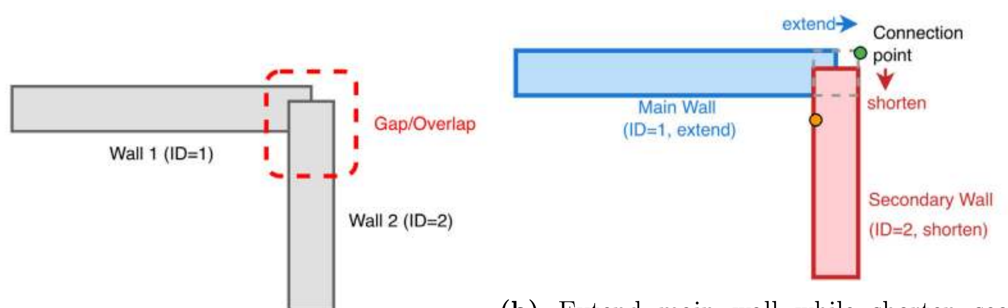
(a) L-shape junction of two walls

(b) T-shape junction of two walls

(c) Endpoint graph of L-shape junction

(d) Endpoint graph of T-shape junction

Fig. 3. Junction type of adjacent walls.



(a) L-shape corner of two walls

(b) Extend main wall while shorten secondary wall

Fig. 4. Endpoint adjustment for L-shape corner.

different levels of reconstruction differ in the IFC entities they generate. The `show_door_window_objects` flag controls whether door and window openings are represented as parametric building objects (`IfcDoor`, `IfcWindow`), or merely as geometric voids (`IfcOpeningElement`) on walls. Similarly, the `create_floors` and `create_ceilings` parameters determine whether horizontal floor and ceiling slabs are generated as `IfcSlab` entities. Moreover, the `create_spaces` parameter controls the generation of `IfcSpace` entities that define volumetric spatial zones bounded by walls, floors, and ceilings. The `create_furniture` parameter decides whether furniture detected in the point cloud is included as `IfcFurnishingElement` entities. Each furniture object has its semantic classification through its type or custom property sets (e.g., "Sofa", "Desk", "Conference Table").

The agent receives user intention from the Query Interpreter Agent. For instance, for a query requesting *walls_only* reconstruction, the exporter generates an IFC file containing exclusively `IfcWall` entities without floors, ceilings, spaces, or furniture, a minimal representation suitable for rapid spatial layout verification. For *basic_room* reconstruction, the exporter includes walls, floors, ceilings, and spaces, with door and window openings represented as `IfcOpeningElement` voids rather than parametric objects. For *full_arch* reconstruction, parametric `IfcDoor` and `IfcWindow` objects are reconstructed with predefined full property sets.

To generate connected geometric primitives, we introduce a wall connection analysis and corner joint calculation algorithm that ensures geometrically precise corner joints in the resulting BIM model, a critical requirement for structural integrity visualization and clash detection. The wall connection detection algorithm starts with constructing an endpoint graph where each wall endpoint becomes a node, and connections are identified through Euclidean distance testing with a configurable tolerance threshold (default 0.01 m). For each wall pair, it checks point-to-point and point-to-line distances to detect whether it is a T- or L-shape junction. The junctions and corresponding endpoint graphs are shown in Fig. 3. For a L-shape corner (shown in Fig. 3a), two walls share the same endpoint node (Endpoint e_2 in Fig. 3c). For a T-shape junction illustrated in Fig. 3b, one endpoint of a wall is connected to the middle rather than two endpoints, as shown in Fig. 3d.

The algorithm then implements geometric adjustments using a hierarchical priority scheme. For a T-shape junction, "main wall" is the wall without any edge point ending at the other wall. In this case, the "secondary wall" is shortened to the inner face of the main wall. For the L-shape corner, for each connection point, the wall with the lower ID is designated as the "main wall" and extends by half the thickness of the intersecting wall to reach the outer corner, while the wall with the higher ID is designated as "secondary" and shortens by half the main wall's thickness to terminate at the inner face. This process is illustrated in Fig. 4. This adjustment strategy produces perfect right-angle corners both on interior and exterior wall surfaces, eliminating the geometric gaps and overlaps in the direct wall-to-IFC conversion approaches. The adjusted wall endpoints are then used to construct extruded solid geometries through `IfcExtrudedAreaSolid` entities.

Algorithm 1 shows the steps of the proposed wall connection detection and geometric adjustment procedure.

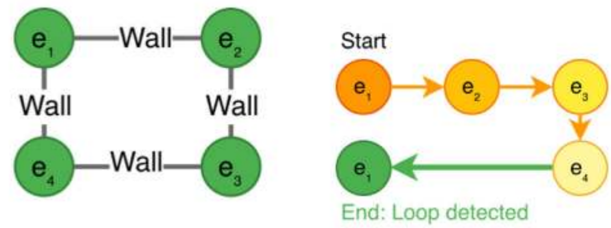
Algorithm 1 Wall Connection Analysis and Corner Adjustment

Require: Walls $W = \{w_1, \dots, w_n\}$ with endpoints and thickness
Ensure: Adjusted wall endpoints for connected IFC geometry

```
Build endpoint graph from all wall endpoints
for each wall pair (w_i, w_j) do
    if endpoint-to-endpoint distance < τ_p then
        Classify as L-shape junction
    else if endpoint-to-centerline distance < τ_l then
        Classify as T-shape junction
    end if
end for
for each detected junction do
    if junction is T-shape then
        Set intersected wall as main wall
        Shorten secondary wall to inner face of main wall
    else if junction is L-shape then
        Set lower-ID wall as main wall
        Extend main wall by half thickness of secondary wall
        Shorten secondary wall by half thickness of main wall
    end if
end for
for each adjusted wall do
    Generate IfcExtrudedAreaSolid
end for
```

Another key feature of the agent is the room detection algorithm that identifies enclosed spatial zones from wall topology without requiring explicit room definitions from the input. The algorithm employs depth-first search on the wall endpoint graph to find closed loops, starting from each connection point and recursively traversing connected walls until returning to the origin. The implementation tracks edges that are already visited to prevent infinite loops and duplicate counting. Fig. 5 shows this process, starting at connection point e_1 (orange) and recursively traversing the endpoint graph through adjacent nodes ($e_1 \rightarrow e_2 \rightarrow e_3 \rightarrow e_4$). When the traversal returns to the origin (e_1, shown in green), completing the closed cycle $e_1 \rightarrow e_2 \rightarrow e_3 \rightarrow e_4 \rightarrow e_1$, a potential room boundary is detected. The color gradient from orange through yellow to green represents the traversal order. Subsequently, the discovered loops are checked by computing the polygon area using the shoelace formula (Gauss's area formula). Polygons with area less than $0.5\ \mathrm{m}^2$ are discarded, and vertex ordering is normalized to counter-clockwise orientation to ensure a positive signed area. This topological room inference enables further generation of floor slabs, ceiling slabs, and space entities. This is particularly useful as the Inference Agent that uses SpatialLM model as a foundation provides only wall primitives, significantly enhancing the completeness of automatically generated BIM models.

Another feature is the translation of room boundaries defined by wall centerlines into floor and ceiling slab footprints that extend to the outer wall faces, resulting in a watertight, cuboid room reconstruction. As illustrated in Fig. 6, for each corner vertex where two walls meet, the algorithm computes the corresponding outer-corner point by intersecting the two outer edge lines of the adjacent walls. The process consists of the following steps: (1) determining the local traversal directions along the room polygon, (2) deriving outward-pointing perpendicular vectors for each wall, (3) offsetting the corner vertex by half the wall thickness along these perpendiculars to form the two outer edge lines, and (4) solving for their intersection to obtain the expanded corner. This geometric expansion ensures that floor and ceiling slabs align with the exterior wall surfaces rather than remaining at centerline positions, producing architecturally correct volumetric representations that are critical for reliable space metrics.



(a) Endpoint graph of room **(b)** Depth-first search from endpoint

Fig. 5. Depth-first search on endpoint graph of room.

This semantic enrichment transforms the output from simple geometric primitives into information-rich BIM files that support a wide range of applications. For instance, energy simulation tools can extract spatial relationships to model heat transfer. Cost estimation systems can query element quantities to generate bills of renovation. Facility management platforms can access space definitions and equipment locations to optimize maintenance schedules. The IFC format makes it possible to use the generated files across different software, making the reconstruction results usable in practical AEC workflows.

3.6. Graph generation agent

Rather than constituting a separate reconstruction objective, the semantic scene graph is generated as a complementary representation of the same reconstructed indoor scene produced for IFC export. The IFC model preserves standards-compliant geometry and BIM semantics for engineering workflows, whereas the graph representation makes entities and their relationships explicitly queryable, which is advantageous for reasoning, retrieval, and graph-based downstream tasks.

Given a layout file produced by the Inference Agent, the Graph Generation Agent constructs a typed, directed graph $G = (V, E)$, where each node $v \in V$ corresponds to a reconstructed entity and each edge $e \in E$ corresponds to a semantic or geometric relationship. The agent supports the layout grammars produced by SpatialLM, including assignment-style statements (e.g., `wall_0=Wall(...)` and `door_0=Door(wall_5, . . . )`). After parsing, nodes have their entity types {*room, wall, door, window, furniture*}. Each node stores a unique identifier, a type label, and other properties. For walls, the properties include start and end coordinates, height, thickness, and a wall length computed from planar endpoints. For doors and windows, the node properties include position, width, height, and an explicit reference to the wall where it is located. For furniture, the node properties store a class label and parameters of the oriented 3D bounding box.

Edges encode both explicit structural and inferred geometric relations. Explicit relations are created directly from the parsed layout semantics, including `has_opening` edges from a wall node to door/window nodes, and `has_wall` edges from a room node to wall nodes. In addition, relations (`contains`) link a room node to furniture nodes. Geometric relations are inferred via endpoint proximity tests: two walls are marked as `adjacent_to` if any pair of their planar endpoints is within a tolerance of threshold $\tau = 0.01$ m.

The current implementation starts with a room node (`Room_0`) when at least one wall is present and attaches all detected walls (and,

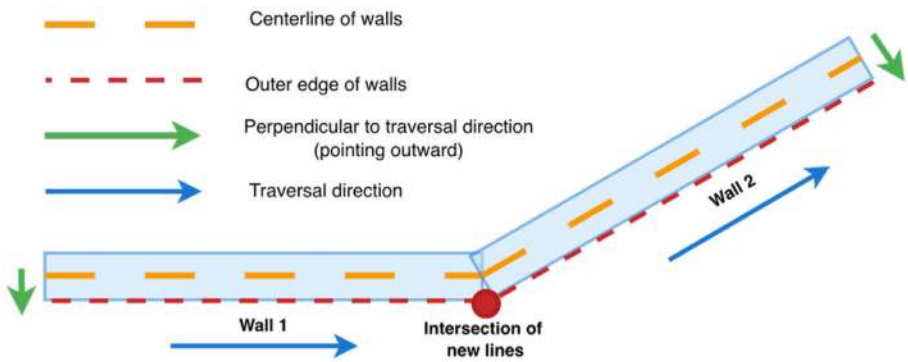


Fig. 6. Compute outer edge for wall centerline.

when available, all furniture) to that room. This conservative heuristic yields a consistent graph structure for single-room scenes and can work as a foundation for more advanced multi-room partitioning methods, such as closed-loop detection and wall-cycle clustering.

Finally, the resulting graph is converted to a JSON document containing a `nodes` array and an `edges` array, where each edge stores `{source, target, relation}`. This representation is straightforward to ingest into graph tooling (e.g., via a simple mapping layer to property-graph databases) and graph learning pipelines, enabling tasks such as semantic retrieval (e.g., retrieving all openings hosted by a given wall), constraint checking, and learning-based scene understanding.

3.7. Validation agent

Unlike the Query Interpreter Agent, which relies on LLM prompting, the Validation Agent does not use LLMs, as validation tasks require deterministic and verifiable outcomes. The Validation Agent provides multi-stage quality assurance throughout the reconstruction pipeline. By checking the outcome at each transformation step, it detects errors early and provides meaningful feedback to users. The agent is executed three times in the whole pipeline.

At the input stage, point cloud validation ensures data integrity before computationally expensive inference. The agent verifies file extension against supported formats (PLY, PCD, XYZ) and attempts to parse the file structure to identify corruption. It then checks that the point cloud contains a sufficient number of points to represent meaningful room scenes, warning users when point clouds are too sparse. Moreover, it also checks the existence of RGB color information, which is important for the reconstruction quality of SpatialLM models.

At the intermediate stage, the Validation Agent assesses the quality of the generated layout code before IFC export using deterministic, rule-based checks rather than LLM-based evaluation. The agent verifies completeness by ensuring that essential elements, such as walls, are present in the layout. It then ensures internal consistency through constraints: geometric parameters (e.g., lengths, heights) must be positive, and semantic references must be valid (e.g., doors and windows must be located on existing walls).

These checks are implemented as programmatic validation rules applied to the structured layout representation, avoiding uncertainty associated with LLM-based judgment. Any detected inconsistencies are reported as errors or warnings and propagated to the Coordinator Agent for handling.

At the output stage, IFC validation uses IfcOpenShell to check schema compliance and geometric validity. The agent verifies that the generated file follows the IFC4 standard, checking entity types, attribute data types, and required relationships. It counts entities of each type (`IfcWall`, `IfcDoor`, `IfcWindow`, `IfcSlab`, `IfcFurnishingElement`, `IfcSpace`) and compares them to values from the layout generation stage. It further validates the spatial hierarchy, ensuring that elements are correctly contained within spaces.

Table 3
Runtime breakdown for a representative scene.

Component	Time (s)
Query interpreter agent	2̃.0
Inference agent (SpatialLM)	2̃0.0
IFC generation	1̃.0
Graph generation	<1.0

All validation outcomes are recorded and propagated to the Coordinator Agent. This provides users with clear explanations of failure modes and improves the reliability, explainability, and practical usability of the automated reconstruction pipeline.

4. Experiments and results

This section presents the experimental setup and evaluation results of the proposed framework. It first describes the implementation details and datasets used, followed by quantitative and qualitative evaluations across different scenarios to assess performance and generalizability.

4.1. Implementation details

The proposed multi-agent framework was evaluated on a workstation equipped with an NVIDIA RTX 4090 GPU and an Intel(R) Xeon(R) Platinum 8470Q CPU, and deployed as a web application. To understand natural-language queries, the Query Interpreter Agent uses the instruction-tuned LLM Qwen2.5 models [59], which converts user requests into structured reconstruction specifications. For geometric inference, the Inference Agent uses SpatialLM1.1-Llama-1B and SpatialLM1.1-Qwen–0.5B [58] to generate room layout code. The peak GPU memory usage during inference was approximately 16 GB when running the Qwen2.5-7B-Instruct foundation model, and around 3 GB when running the SpatialLM1.1-Llama-1B model.

To assess the practical applicability of the proposed framework, we analyze its computational performance in terms of runtime. Table 3 presents a breakdown of processing time for each component of the framework on a representative scene (excluding time for visualizing and rendering IFC or graphs). The results show that the overall computational cost is dominated by the Inference Agent, while IFC generation, graph construction, and validation introduce relatively small costs.

The current framework operates in a batch-processing mode, where each scene is processed independently. The modular design allows different agents to be executed sequentially or in parallel across multiple scenes, enabling efficient batch processing in large-scale workflows. While streaming or incremental processing is not explicitly implemented in this work, the use of structured intermediate representations

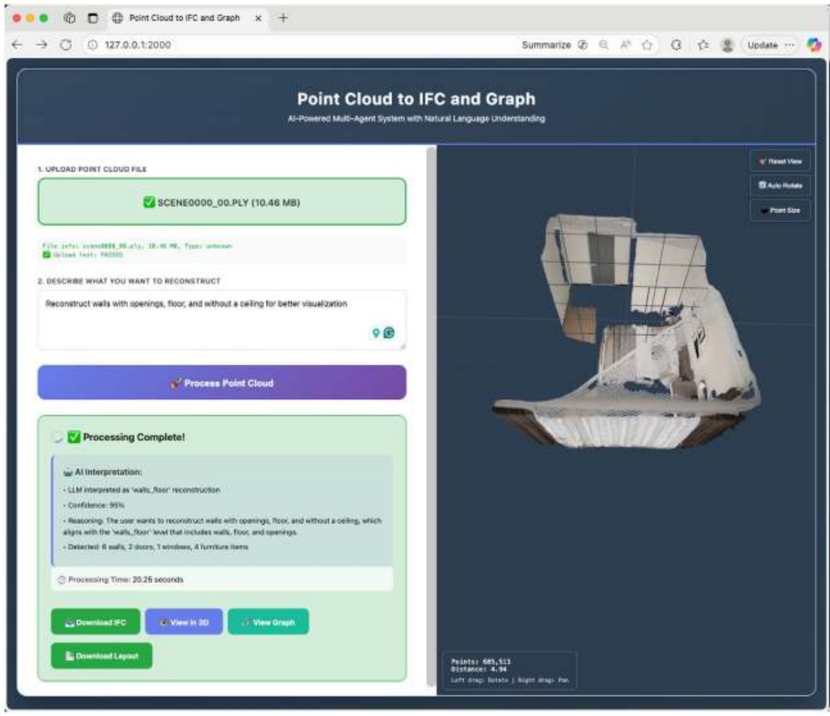


Fig. 7. User interface of developed framework.

(e.g., layout code and semantic graphs) provides a basis for future extensions.

The User Interface (UI), shown in Fig. 7, allows users to upload a point cloud file and specify a reconstruction query in natural language, while providing an interactive visualization of the point cloud data. After processing, the application returns the reconstructed results for download, including the generated IFC model and the room-layout file.

In addition, the application supports inspection of the outputs via an IFC viewer and a semantic graph viewer (Fig. 8), enabling users to verify the reconstructed geometry and explore the extracted relationships without leaving the application.

To assess generalizability across sensing modalities, we tested the framework on point clouds from three datasets: (a) the TUMCMS dataset (together with manually created BIM model as ground truth [60]), acquired in office environments using mobile laser scanning; (b) S3DIS [61], captured in indoor office buildings with an RGB-D camera system; and (c) the synthetic SpatialLM dataset [62].

It should be noted that, although the datasets are acquired using different sensing modalities (e.g., mobile laser scanning and RGB-D), all inputs to the framework are point clouds. Therefore, the evaluation focuses on the framework's ability to generalize across point clouds with different characteristics, such as noise patterns, density, and completeness, rather than across raw sensor modalities directly.

4.2. Results and discussion

In this section, we show and evaluate the performance of the proposed framework.

4.2.1. Evaluation of query interpretation

To evaluate the performance of the Query Interpreter Agent, we construct a set of 25 natural-language queries covering different reconstruction intents. Each query is associated with a ground truth reconstruction level from the predefined set {*walls_only, walls_floor, basic_room, full_arch, with_furniture*}. The full set of evaluation questions is provided in Appendix (Table A.9).

Table 4
Performance of query interpretation.

Model	Accuracy
Qwen2.5-0.5B-Instruct	60%
Qwen2.5-1.5B-Instruct	76%
Qwen2.5-3B-Instruct	84%
Qwen2.5-7B-Instruct	96%

We evaluate multiple instruction-tuned LLMs and measure their accuracy in mapping queries to the correct reconstruction configuration. Table 4 summarizes the results. The results show that the 7B model achieves the highest accuracy (96%), followed by the 3.5B model (84%) and 1.5B model (76%), while the 0.5B model performs significantly worse.

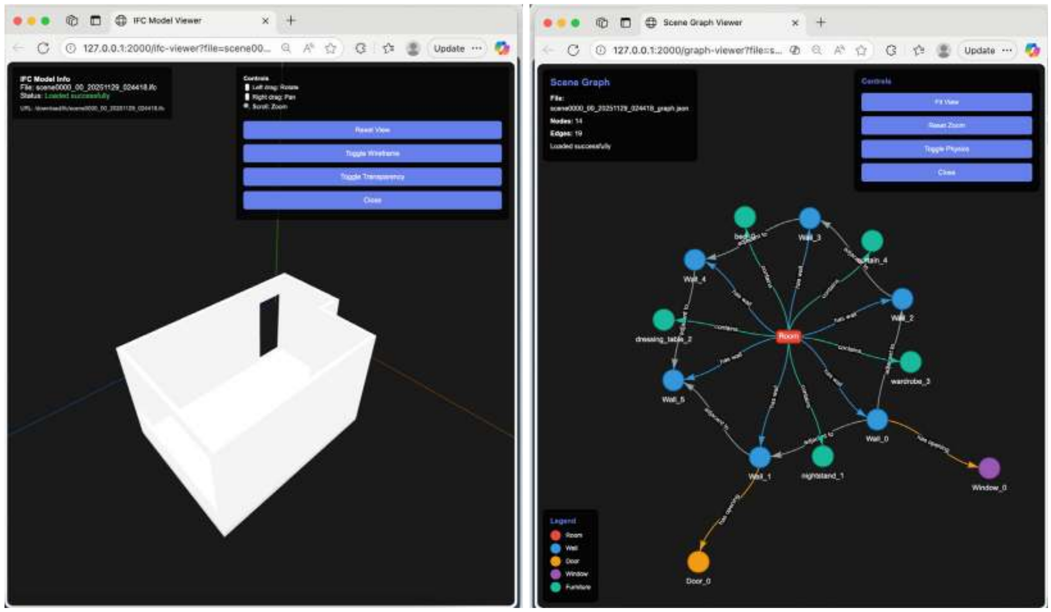
The results indicate that the Query Interpreter Agent is effective in mapping natural-language queries to structured reconstruction configurations, particularly when the intent is clearly aligned with predefined reconstruction levels. The best-performing model achieves an accuracy of 96%, demonstrating that lightweight LLMs can support intent-driven configuration with relatively good reliability.

4.2.2. TUMCMS dataset

We evaluate the proposed framework on the TUMCMS dataset and report both qualitative and quantitative results on this dataset.

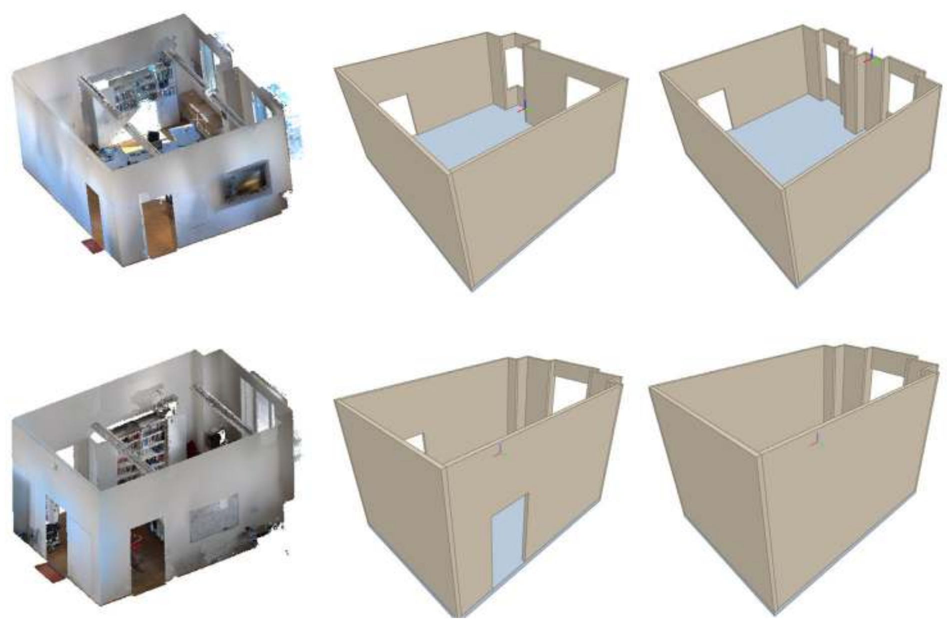
Fig. 9 presents representative qualitative reconstructions for two office scenes (ceiling points/slabs are removed for better visualization). The input point clouds are shown in Fig. 9a. Fig. 9b shows the corresponding IFC models generated using SpatialLM1.1-Llama-1B as the foundation model, while Fig. 9c shows the IFC outputs produced using SpatialLM1.1-Qwen–0.5B.

Overall, both models reconstruct the room layout geometry reliably, generating correct wall connections and enclosed spaces. Notably, in these two examples, the Qwen-based model captures complex room boundaries more accurately, particularly for rooms in irregular shapes with multiple short, connected wall segments. In contrast, openings



(a) IFC viewer (b) Graph viewer

Fig. 8. IFC and graph viewer of framework.



(a) Example of point cloud scenes (b) Generated by LLama-based model (c) Generated by Qwen-based model

Fig. 9. Examples of point cloud and generated IFC files in TUMCMS dataset (Ceiling is removed for better visualization).

are less consistently reconstructed. For windows, both models detect most window openings, whereas door opening detection performs worse. Several doors are missed by both models, with SpatialLM1.1-Qwen–0.5B showing a higher miss rate in these examples.

This behavior is expected given the domain gap between training and evaluation data [63]. Both foundation models are trained on the synthetic SpatialLM dataset [62], which encodes different sensor characteristics and appearance cues compared to laser-scanned point clouds in TUMCMS. In the TUMCMS dataset, walls show similar visual cues, while windows often contain reflections or noise points due to glass surfaces. Moreover, doors in a room scene may appear either as a complete door instance with geometry or as void openings. If the training distribution mainly contains door instances with visible geometry, models may struggle to recognize doors represented mainly as empty openings in the wall. These discrepancies explain the strong performance on wall layout reconstruction and the reduced robustness for door and window openings in the laser-scanned dataset.

For quantitative evaluation, room area is used as the quantitative metric because it reflects the correctness of the reconstructed spatial layout, which is directly determined by wall positions and connectivity. Errors in wall placement, wall alignment, and corner junction handling will cause deviations in the room area. In addition, room area is a standard and practically relevant quantity in BIM-based workflows, as it is widely used in applications such as space management, cost

Table 5
Area comparison between reconstruction of framework and BIM model on TUMCMS test set.

Office no.	Foundation	Framework (m^2)	BIM (m^2)	Abs. Dev. (m^2)	Rel. Dev. (%)
Office 1	Llama-based	52.43	49.22	3.21	6.52
Office 2	Llama-based	28.10	26.82	1.28	4.77
Office 3	Llama-based	25.52	24.11	1.41	5.85
Office 4	Llama-based	26.97	25.88	1.09	4.21
Office 5	Llama-based	25.83	25.20	0.63	2.50
Office 6	Llama-based	25.98	24.90	1.08	4.34
Average (Llama-based)				**1.45**	**4.70**
Office 1	Qwen-based	51.77	49.22	2.55	5.18
Office 2	Qwen-based	28.82	26.82	2.00	7.46
Office 3	Qwen-based	25.48	24.11	1.37	5.68
Office 4	Qwen-based	27.68	25.88	1.80	6.96
Office 5	Qwen-based	26.11	25.20	0.91	3.61
Office 6	Qwen-based	25.32	24.90	0.42	1.69
Average (Qwen-based)				**1.51**	**5.10**

Table 6
Comparison of area reconstruction accuracy across methods on TUMCMS test set.

Method	Abs. Dev. (m^2)		Rel. Dev. (%)	
	Mean	Std	Mean	Std
Void-growing (5 cm)	1.23	0.71	4.02	1.41
Void-growing (3 cm)	0.87	0.43	2.96	1.39
Llama-based	1.45	0.89	4.70	1.48
Qwen-based	1.51	0.74	5.10	2.04

Table 7
Detection error rates for doors and windows.

Element	Model	Missing rate	Over-detection rate
Doors	LLaMA	8.3%	0.0%
	Qwen	16.7%	8.3%
Windows	LLaMA	12.5%	0.0%
	Qwen	25.0%	12.5%

estimation, and energy analysis. Therefore, evaluating area consistency provides both a geometric validation and an application-oriented assessment of reconstruction quality.

Table 5 reports the room areas extracted from *IFCQUANTITYAREA* for the reconstructed IFC models and compares them against areas from the manually reconstructed BIM. Across these offices, the framework using both foundation models presents results close to ground truth, with average deviations around 1.5 m^2 and 5%.

Table 6 compares the proposed framework against a previous geometric baseline (void-growing) under two voxel resolutions [60]. Overall, the void-growing approach achieves the lowest errors on this test set, and its accuracy improves as the resolution increases from 5 cm to 3 cm. In comparison, the Llama-based and Qwen-based approaches of our framework achieve mean absolute deviations of 1.45 m^2 (std 0.89) and 1.51 m^2 (std 0.74), respectively, with mean relative deviations of 4.70% (std 1.48) and 5.10% (std 2.04). While these approaches do not perform better than the void growing approach, the proposed method remains comparable and still competitive with the 5 cm configuration.

These results are acceptable given that the proposed framework is not optimized for the TUMCMS dataset. In contrast to the void-growing baseline, our framework is evaluated in a strictly zero-shot way. We do not train, fine-tune, or adapt any component using the TUMCMS training set. Instead, the entire pipeline relies on foundation models trained on synthetic data and is applied directly to laser-scanned TUMCMS point clouds. Without dataset-specific adaptation, the framework still achieves competitive area accuracy, showing its generalizability in reconstructing IFC files across datasets and sensors.

4.2.3. SpatialLM dataset

We further evaluate the framework on the synthetic SpatialLM dataset to assess performance under challenging indoor scenes. Fig. 10 shows representative qualitative results for three scenes (ceiling slabs are omitted for clarity). The input point clouds are presented in Fig. 10a, while the reconstructed IFC models produced by SpatialLM1.1-Llama-1B and SpatialLM1.1-Qwen–0.5B are shown in Fig. 10b and Fig. 10c, respectively.

The SpatialLM test set contains particularly difficult samples with substantial noise, clutter, and occlusions. The framework with both foundation models can generate IFC outputs with correct wall networks and enclosed room spaces, showing its robustness to incomplete input geometry. Differences between reconstructions are most apparent in heavily occluded regions where large portions of the scene are missing. In such cases, the inferred layouts may differ. It should be noted that this behavior is expected, as even manual annotation becomes ambiguous when required boundaries are hardly observed in the point cloud.

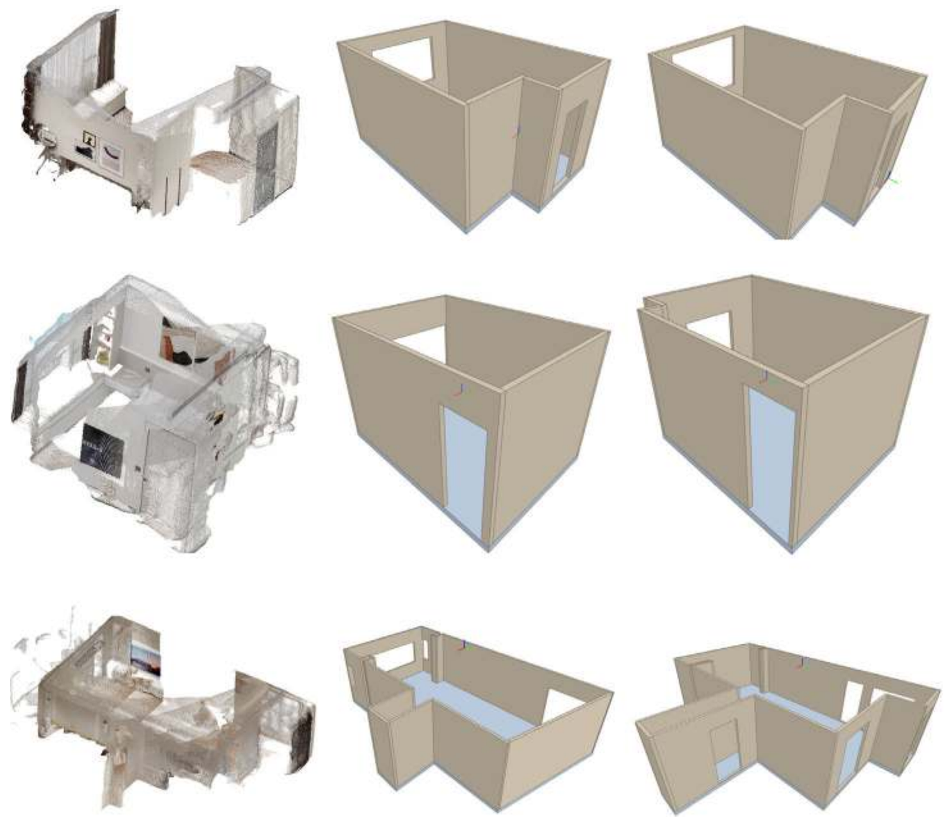
4.2.4. S3DIS dataset

We additionally evaluate the framework on the S3DIS dataset, which is collected using an RGB-D sensing system and shows different noise characteristics and occlusion patterns compared with laser-scanned or synthetic data. Fig. 11 presents representative qualitative results for two scenes (ceiling points/slabs are removed for better visualization). The input point clouds are shown in Fig. 11a, while the reconstructed IFC models generated with SpatialLM1.1-Llama-1B and SpatialLM1.1-Qwen–0.5B are shown in Fig. 11b and Fig. 11c, respectively.

For these examples, the framework with both foundation models can generate almost consistent wall layouts, particularly in the second scene, where the wall topology is complex. In addition, most door and window openings are reconstructed, proving that the framework can infer openings under RGB-D sensing conditions. Nevertheless, failures remain in some cases. For instance, glass doors are occasionally missed due to weak or noisy geometric patterns. In our examples, the Llama-based reconstruction misses one door on the long wall that is recovered by the Qwen-based model in the second scene. Moreover, the dimensions of these openings are not always consistent in different reconstructions.

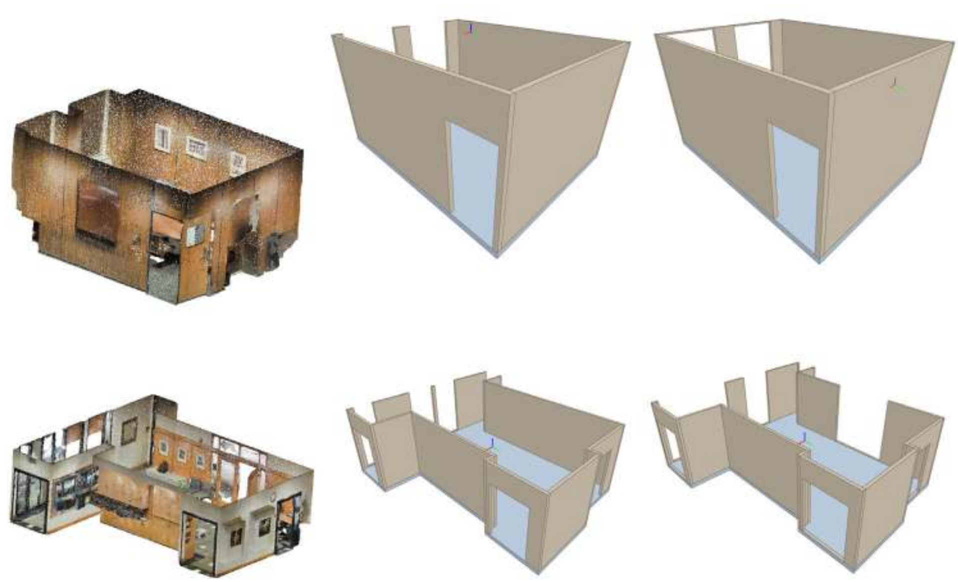
Table 7 presents detection performance in terms of missing and over-detection rates for five randomly selected rooms from the S3DIS dataset. The missing rate and over-detection rate are computed based on differences between predicted and ground truth element counts, providing a normalized measure of under- and over-detection across rooms.

The LLaMA-based model consistently achieves lower error rates, with no over-detection and relatively low missing rates for both doors



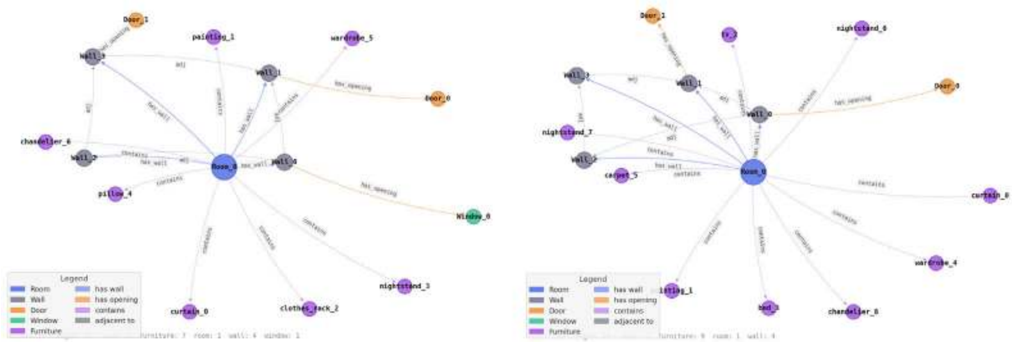
(a) Example of point cloud scenes (b) Generated by LLama-based model (c) Generated by Qwen-based model

Fig. 10. Examples of point cloud and generated IFC files in SpatialLM dataset (Ceiling is removed for better visualization).

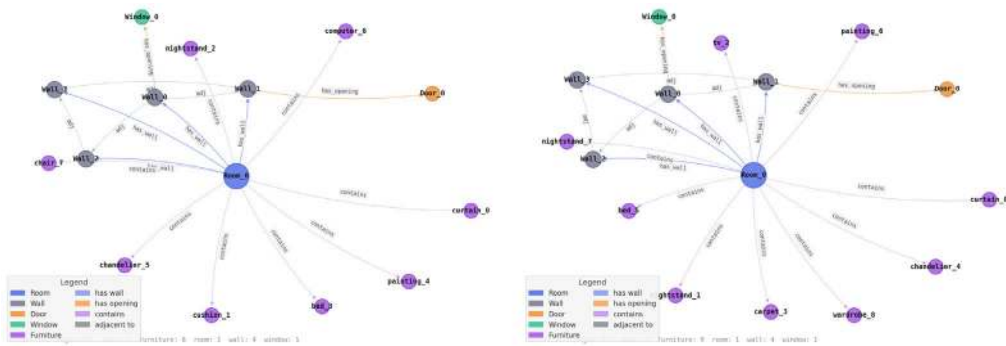


(a) Example of point cloud scenes (b) Generated by LLama-based model (c) Generated by Qwen-based model

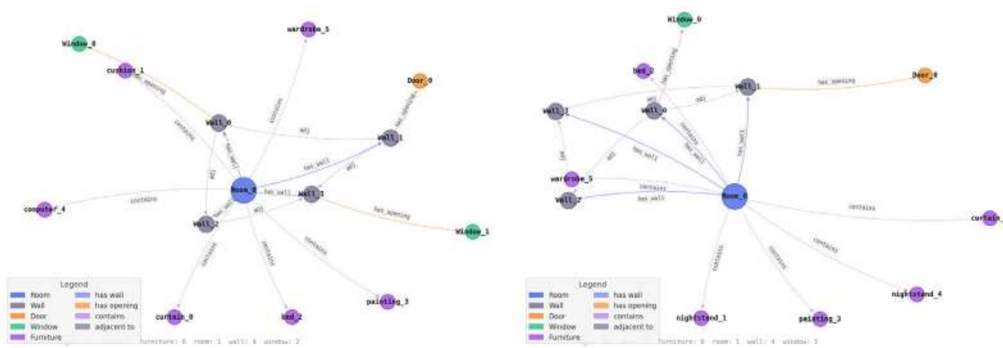
Fig. 11. Examples of point cloud and generated IFC files in S3DIS dataset (Ceiling is removed for better visualization).



(a) Ground truth of semantic graphs



(b) Graphs generated from Llama-based predictions



(c) Graphs generated from Qwen-based predictions

Fig. 12. Examples comparing ground truth graphs with the generated graphs.

and windows. In contrast, the Qwen-based model exhibits higher missing rates and occasional over-detection, indicating less stable performance. These results suggest that while both models can capture the overall presence of doors/windows, LLaMA provides more reliable and balanced predictions. The errors are generally small, reinforcing that the framework can maintain semantic consistency, although precise instance-level detection remains an area for improvement.

The results on the S3DIS dataset further demonstrate the generalizability of the proposed framework. Without any training or fine-tuning on S3DIS, it reconstructs high-quality wall layouts in IFC models.

4.2.5. Semantic graph evaluation

The semantic graph generation in the proposed framework is deterministic, as graph nodes and relationships are directly derived from the structured layout representation produced by the Inference Agent. Therefore, the correctness of the generated graph depends on the consistency of the transformation process.

To assess the quality of the generated graphs, we perform consistency validation between the graph representation and the underlying layout. Specifically, the following aspects are verified: (a) all detected entities (e.g., walls, rooms, openings, and furniture) are correctly represented as graph nodes; (b) relationships such as *adjacent_to* and *contains* are consistent with geometric and semantic definitions derived from the layout, and (c) no invalid or dangling references exist in the graph.

Across all evaluated scenes, the generated graphs are fully consistent with the corresponding layout representations, with no missing nodes or invalid relationships observed. Fig. 12 illustrates some examples comparing ground truth graphs with the generated graphs. The results show that key spatial relationships, including wall connectivity and containment hierarchies, are correctly captured. It should be noted that any potential errors in the semantic graph are primarily propagated from detection results in previous steps (e.g., missed or incorrect element identification), rather than from the graph generation process itself.

Table 8
Comparison between conventional configuration and LLM-based query interface.

Aspect	Conventional approach	Proposed approach
Input format	Parameter settings/config files	Natural-language query
User requirement	Knowledge of pipeline and schema	High-level intent specification
Flexibility	Limited to predefined options	Flexible and expressive queries
Workflow steps	Multiple	Single query
Manual correction	Frequent	Reduced

The graph generation process operates on structured layout representations and scales linearly with the number of elements in the scene. While this study focuses on room-level reconstruction, the approach can be extended to larger environments by aggregating room-level graphs into higher-level structures (e.g., floor- or building-level graphs). Evaluating large-scale graph consistency remains an important direction for future work.

To summarize, the proposed multi-agent framework demonstrates strong generalizability and robust performance in different indoor point cloud datasets and sensing modalities. All reported experiments are conducted in a zero-shot setting: no training, fine-tuning, or dataset-specific parameter adaptation is performed on TUMCMS, SpatialLM, or S3DIS. Despite this, the framework consistently produces high-quality IFC models, with particularly reliable reconstruction of wall topology and enclosed room spaces. Quantitatively, the inferred room areas are also close to the manually created BIM model.

4.3. Discussion

In comparison with conventional configuration approaches, traditional Scan-to-BIM workflows typically require users to specify reconstruction parameters explicitly, such as element types, levels of detail, often through predefined interfaces or configuration files. This requires familiarity with both the reconstruction pipeline and BIM schema definitions.

In contrast, the proposed framework enables an intent-driven configuration mechanism, where reconstruction requirements are expressed in natural language and translated into structured configurations by the Query Interpreter Agent. This allows users to specify high-level goals (e.g., reconstruction scope and level of detail) without directly interacting with low-level parameters.

Table 8 shows a comparison between a conventional parameter-based workflow and the proposed approach. The proposed approach provides a more flexible and expressive way to define reconstruction tasks, while maintaining compatibility with structured downstream processing.

Furthermore, the proposed framework reduces the need for manual configuration, eliminating the requirement for explicit parameter tuning and step-by-step setup commonly found in conventional workflows. In addition, the topology-aware IFC generation and deterministic graph construction reduce the need for manual correction of geometric inconsistencies and missing relationships. The reduction in manual intervention is reflected in the simplified workflow, where reconstruction can be performed through a single high-level query rather than multiple configuration steps. A systematic evaluation of manual effort, including correction time and user interaction, remains an important direction for future work.

In addition, the current framework is primarily designed for indoor environments with relatively regular layouts. As a result, it may face challenges in handling more complex geometries, such as non-Manhattan wall configurations, varying wall thicknesses, irregular junctions, and partially enclosed spaces. The geometric abstraction and rule-based processing adopted in this work assume planar elements, which may not fully capture such irregularities.

The observed difficulty in detecting windows and doors in the TUMCMS dataset shows the impact of sensor differences between training and testing data. These domain gaps, arising from different sensing modalities [64], can lead to inconsistent detection results. Improving cross-dataset generalization, for example, through domain adaptation or multi-dataset training, remains an important direction for future work.

In the proposed framework, generalizability is primarily influenced by the underlying detection model (e.g., SpatialLM), rather than the downstream IFC and graph generation processes. Other failure modes, apart from over- or misdetected openings (doors and windows), include inaccurate wall boundaries that lead to incorrect room topology, as well as ambiguities in cluttered or partially enclosed spaces. These errors may propagate to both the generated IFC models and semantic graphs. The framework partially mitigates this limitation through its modular design. Specifically, the separation of detection, representation, and reconstruction allows the system to adapt to different datasets by replacing or fine-tuning the detection component without modifying the downstream pipeline.

5. Conclusion

This paper presented an LLM-enabled multi-agent framework that automates the conversion of indoor point clouds into standards-compliant IFC BIM models and corresponding semantic graphs. By decomposing the Scan-to-BIM and Scan-to-Graph workflow into specialized agents for query interpretation, geometric inference, IFC generation, graph construction, and multi-stage validation, the proposed framework achieves reliable and robust performance. Experimental results across three datasets show that the framework can reconstruct wall layouts and enclosed spaces, and the room area estimation is comparable to a manually created BIM model.

From a practical perspective, the framework reduces the need for manual configuration typically required in Scan-to-BIM and Scan-to-Graph workflows, which often require extensive post-processing and expert knowledge.

The use of LLM-based query interpretation reduces the knowledge barrier for end users by allowing reconstruction goals and output requirements to be specified in natural language rather than through specific parameters. Furthermore, all experiments were conducted in a zero-shot setting without training or fine-tuning on corresponding training sets, and the framework still achieved robust performance across datasets captured with different sensors (like laser scanning and RGB-D), showing its potential for scalable deployment in various real-world environments.

However, this study still has several limitations. Fully automatic reconstruction can introduce errors, implying a trade-off between the level of automation and the degree of human oversight required for quality assurance. In addition, a typical problem is performance degradation in scenes with heavy clutter, strong noise, or severe occlusions. Furthermore, domain gaps between sensing modalities affect the reconstruction of some elements, such as door and window openings, where representation differences (complete door geometry versus only void openings) can lead to missed detections or inconsistent dimensions.

In the future, we will focus on improving robustness and consistency across sensor types and environmental conditions. A promising direction is to develop more comprehensive LLM-integrated scene understanding models that achieve reasoning over geometry, appearance, and building knowledge. Adding uncertainty estimation and optional human-in-the-loop correction mechanisms are another important step towards deploying automated scan-to-BIM and Scan-to-Graph pipelines in workflows that are critical for safety and compliance.

Table A.9
Natural-language queries used for evaluating the Query Interpreter Agent.

ID	Query	Ground truth
Q1	Just give me the walls, nothing else.	walls_only
Q2	I only need the wall structure for this layout.	walls_only
Q3	Export walls only, no floors or ceilings.	walls_only
Q4	I need to check room adjacency — keep it minimal.	walls_only
Q5	This is for a fire egress analysis, just the partitions.	walls_only
Q6	Give me walls and floors, no ceiling.	walls_floor
Q7	I need a floor plan with slabs but skip the ceiling.	walls_floor
Q8	Reconstruct the floor and walls without any openings.	walls_floor
Q9	I want to visualise the room extents without occlusion from above.	walls_floor
Q10	This is for area calculation — I need the floor surfaces and bounding walls.	walls_floor
Q11	Give me a full room with walls, floor, ceiling, and opening voids.	basic_room
Q12	Complete room geometry but I don't need door or window objects, voids are fine.	basic_room
Q13	Walls, floors, ceilings and spaces — openings as holes, not objects.	basic_room
Q14	I need a structural room model for CFD simulation, including openings.	basic_room
Q15	Quick preview for a client meeting — full enclosure with openings, no details.	basic_room
Q16	Full architecture with door and window objects, no furniture.	full_arch
Q17	Create an architectural BIM model with doors and windows.	full_arch
Q18	Walls, floors, ceilings, parametric doors and windows.	full_arch
Q19	I need this for a renovation quote — contractor needs to see windows and doors.	full_arch
Q20	Prepare the model for IFC export to Revit, include architectural elements.	full_arch
Q21	Give me everything — walls, floors, doors, windows, and furniture.	with_furniture
Q22	Full scene reconstruction including furniture and fixtures.	with_furniture
Q23	Complete model with furniture for an interior design presentation.	with_furniture
Q24	We need the complete digital twin including all movable objects.	with_furniture
Q25	Make it as detailed as possible.	with_furniture

CRediT authorship contribution statement

Yuandong Pan: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Mudan Wang:** Writing – review & editing, Writing – original draft, Methodology, Investigation, Conceptualization. **Jiechao Gao:** Writing – review & editing, Resources, Methodology, Conceptualization. **Jie Wang:** Writing – review & editing, Supervision, Project administration, Conceptualization. **Michael D. Lepech:** Writing – review & editing, Supervision, Resources, Project administration. **Ioannis Brilakis:** Writing – review & editing, Supervision, Resources, Project administration, Funding acquisition.

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.

Acknowledgments

This project was supported by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 101034337. In addition, this project has received funding from the Horizon Europe UKRI Underwrite Innovate, under the grant G115676.

Appendix. Query set for evaluation of Query Interpreter Agent

See Table A.9.

Data availability

Data will be made available on request.

References

[1] I. Brilakis, Y. Pan, A. Borrmann, H.-G. Mayer, F. Rhein, C. Vos, E. Pettinato, S. Wagner, Built environment digital twining, in: International Workshop on Built Environment Digital Twinning presented by TUM IAS and Siemens AG, 2019, http://dx.doi.org/10.17863/CAM.65445.

[2] M. Wang, C.C. Wang, S. Sepasgozar, S. Zlatanova, A systematic review of digital technology adoption in off-site construction: Current status and future direction towards industry 4.0, Buildings 10 (11) (2020) http://dx.doi.org/10.3390/buildings10110204.

[3] J. Zhu, N. Nisbet, R. Kang, Y. Wen, M. Wang, I. Brilakis, Revealing the internal structure of IFC-graph for efficient querying and knowledge discovery, Adv. Eng. Inform. 70 (2026) 104229, http://dx.doi.org/10.1016/j.aei.2025.104229, URL https://www.sciencedirect.com/science/article/pii/S147403462501122X.

[4] Y. Pan, M. Wang, L. Lu, R. Wei, S. Cavazzi, M. Peck, I. Brilakis, Scan-to-graph: Automatic generation and representation of highway geometric digital twins from point cloud data, Autom. Constr. 166 (2024) 105654, http://dx.doi.org/10.1016/j.autcon.2024.105654, URL https://www.sciencedirect.com/science/article/pii/S092658052400390X.

[5] Q. Wang, J. Guo, M.-K. Kim, An application oriented scan-to-BIM framework, Remote. Sens. 11 (3) (2019) 365, URL https://www.mdpi.com/2072-4292/11/3/365.

[6] X. Liang, Z. Fu, C. Sun, Y. Hu, MHIBS-Net: Multiscale hierarchical network for indoor building structure point clouds semantic segmentation, Int. J. Appl. Earth Obs. Geoinf. 102 (2021) 102449, http://dx.doi.org/10.1016/j.jag.2021.102449, URL https://www.sciencedirect.com/science/article/pii/S0303243421001562.

[7] M. Wang, C.C. Wang, S. Zlatanova, X. Shen, I. Brilakis, A streamlined laser scanning verticality check method for installation of prefabricated wall panels, J. Constr. Eng. Manag. 150 (11) (2024) http://dx.doi.org/10.1061/JCEMD4.COENG-14989.

[8] X. Xiong, A. Adan, B. Akinci, D. Huber, Automatic creation of semantically rich 3D building models from laser scanner data, Autom. Constr. 31 (2013) 325–337, http://dx.doi.org/10.1016/j.autcon.2012.10.006, URL https://www.sciencedirect.com/science/article/pii/S0926580512001732.

[9] F.C. Collins, S. Mafipour, F. Noichl, Y. Pan, M.A. Vega, M. Disser, A. Hoffmann, L. Kuhn, P. Scheich, Towards applicable scan-to-BIM and scan-to-floorplan: an end-to-end experiment, in: 32. Forum Bauinformatik 2021, Institut für Numerische Methoden und Informatik im Bauwesen, TU Darmstadt, 2021, pp. 98–105, URL https://publications.cms.bgu.tum.de/DTG_STB_FBI.pdf.

[10] W. Shi, W. Ahmed, N. Li, W. Fan, H. Xiang, M. Wang, Semantic geometric modelling of unstructured indoor point cloud, ISPRS Int. J. Geo-Information 8 (1) (2019) 9, URL https://www.mdpi.com/2220-9964/8/1/9.

[11] D. Chen, L. Wan, F. Hu, J. Li, Y. Chen, Y. Shen, J. Peethambaran, Semantic-aware room-level indoor modeling from point clouds, Int. J. Appl. Earth Obs. Geoinf. 127 (2024) 103685, http://dx.doi.org/10.1016/j.jag.2024.103685, URL https://www.sciencedirect.com/science/article/pii/S1569843224000396.

[12] X. Gao, R. Yang, J. Tan, Y. Liu, Floor plan reconstruction from indoor 3D point clouds using iterative RANSAC line segmentation, J. Build. Eng. 89 (2024) 109238, http://dx.doi.org/10.1016/j.jobe.2024.109238, URL https://www.sciencedirect.com/science/article/pii/S2352710224008064.

[13] Y. Pan, A. Braun, A. Borrmann, I. Brilakis, Void-growing: a novel scan-to-BIM method for manhattan world buildings from point cloud, in: Proceedings of the 2021 European Conference on Computing in Construction, in: Computing in Construction, vol. 2, University College Dublin, Online, 2021, pp. 312–321, http://dx.doi.org/10.35490/ec3.2021.162.

[14] J. Wei, H. Wu, H. Yue, S. Jia, J. Li, C. Liu, Automatic extraction and reconstruction of a 3D wireframe of an indoor scene from semantic point clouds, Int. J. Digit. Earth 16 (1) (2023) 3239–3267, http://dx.doi.org/10.1080/17538947.2023.2246937, arXiv:https://doi.org/10.1080/17538947.2023.2246937.

[15] J. Jung, C. Stachniss, S. Ju, J. Heo, Automated 3D volumetric reconstruction of multiple-room building interiors for as-built BIM, Adv. Eng. Inform. 38 (2018) 811–825, http://dx.doi.org/10.1016/j.aei.2018.10.007, URL https://www.sciencedirect.com/science/article/pii/S1474034618300600.

[16] M. Mahmoud, W. Chen, Y. Yang, Y. Li, Automated BIM generation for large-scale indoor complex environments based on deep learning, Autom. Constr. 162 (2024) 105376, http://dx.doi.org/10.1016/j.autcon.2024.105376, URL https://www.sciencedirect.com/science/article/pii/S0926580524001122.

[17] H. Son, C. Kim, Semantic as-built 3D modeling of structural elements of buildings based on local concavity and convexity, Adv. Eng. Inform. 34 (2017) 114–124, http://dx.doi.org/10.1016/j.aei.2017.10.001, URL https://www.sciencedirect.com/science/article/pii/S1474034617302124.

[18] F. Noichl, Y. Pan, A. Borrmann, Automated steel structure model reconstruction through point cloud instance segmentation and parametric shape fitting, J. Inf. Technol. Constr. 30 (2025) 1099–1122, http://dx.doi.org/10.36680/j.itcon.2025.045.

[19] Y. Pan, F. Noichl, A. Braun, A. Borrmann, I. Brilakis, Automatic creation and enrichment of 3D models for pipe systems by co-registration of laser-scanned point clouds and photos, 2022, http://dx.doi.org/10.17863/CAM.111238, URL https://www.repository.cam.ac.uk/handle/1810/372376.

[20] F. Noichl, Y. Pan, M.S. Mafipour, A. Braun, I. Brilakis, A. Borrmann, From enriched point cloud to structural and MEP models: An automated approach to create semantic-geometric models for industrial facilities, in: Computing in Civil Engineering 2023, pp. 92–99, http://dx.doi.org/10.1061/9780784485224.012.

[21] B. Wang, C. Yin, H. Luo, J.C.P. Cheng, Q. Wang, Fully automated generation of parametric BIM for MEP scenes based on terrestrial laser scanning data, Autom. Constr. 125 (2021) 103615, http://dx.doi.org/10.1016/j.autcon.2021.103615, URL https://www.sciencedirect.com/science/article/pii/S0926580521000662.

[22] A. Dimitrov, M. Golparvar-Fard, Segmentation of building point cloud models including detailed architectural/structural features and MEP systems, Autom. Constr. 51 (2015) 32–45, http://dx.doi.org/10.1016/j.autcon.2014.12.015, URL https://www.sciencedirect.com/science/article/pii/S0926580514002593.

[23] X. Chen, Y. Pan, V.J. Gan, K. Yan, 3D reconstruction of semantic-rich digital twins for ACMV monitoring and anomaly detection via scan-to-BIM and time-series data integration, Dev. Built Environ. 19 (2024) 100503, http://dx.doi.org/10.1016/j.dibe.2024.100503, URL https://www.sciencedirect.com/science/article/pii/S2666165924001844.

[24] F. Xue, W. Lu, K. Chen, Automatic generation of semantically rich as-built building information models using 2D images: A derivative-free optimization approach, Comput.-Aided Civ. Infrastruct. Eng. 33 (11) (2018) 926–942, http://dx.doi.org/10.1061/(ASCE)CP.1943-5487.0000839.

[25] Y. Pan, A. Braun, I. Brilakis, A. Borrmann, Enriching geometric digital twins of buildings with small objects by fusing laser scanning and AI-based image recognition, Autom. Constr. 140 (2022) 104375, http://dx.doi.org/10.1016/j.autcon.2022.104375.

[26] J.W. Ma, T. Czerniawski, F. Leite, Semantic segmentation of point clouds of building interiors with deep learning: Augmenting training datasets with synthetic BIM-based point clouds, Autom. Constr. 113 (2020) 103144, http://dx.doi.org/10.1016/j.autcon.2020.103144, URL https://www.sciencedirect.com/science/article/pii/S0926580519311884.

[27] S. Zhou, J.-R. Lin, P. Pan, Y. Pan, I. Brilakis, Impact of color and mixing proportion of synthetic point clouds on semantic segmentation, Autom. Constr. 171 (2025) 105963, http://dx.doi.org/10.1016/j.autcon.2025.105963, URL https://www.sciencedirect.com/science/article/pii/S0926580525000032.

[28] H. Alavi, N. Forcada, R. Bortolini, D.J. Edwards, Enhancing occupants' comfort through BIM-based probabilistic approach, Autom. Constr. 123 (2021) 103528, http://dx.doi.org/10.1016/j.autcon.2020.103528, URL https://www.sciencedirect.com/science/article/pii/S0926580520311080.

[29] R. Sacks, I. Brilakis, E. Pikas, H.S. Xie, M. Girolami, Construction with digital twin information systems, Data-Cent. Eng. 1 (2020) e14, http://dx.doi.org/10.1017/dce.2020.16.

[30] W. Saif, S. RazaviAlavi, M. Kassem, Construction digital twin: a taxonomy and analysis of the application-technology-data triad, Autom. Constr. 167 (2024) 105715, http://dx.doi.org/10.1016/j.autcon.2024.105715, URL https://www.sciencedirect.com/science/article/pii/S0926580524004515.

[31] M. Wang, Y. Pan, L. Lu, E. Pärn, J. Liu, I. Brilakis, Automating graph-based geometric digital model generation for building digital twin applications from point cloud and image data, Build. Environ. 291 (2026) 114211, http://dx.doi.org/10.1016/j.buildenv.2026.114211, URL https://www.sciencedirect.com/science/article/pii/S036013232600017X.

[32] F. Pfitzner, S. Hu, A. Braun, A. Borrmann, Y. Fang, Monitoring concrete pouring progress using knowledge graph-enhanced computer vision, Autom. Constr. 174 (2025) 106117, http://dx.doi.org/10.1016/j.autcon.2025.106117, URL https://www.sciencedirect.com/science/article/pii/S0926580525001578.

[33] Z. Wang, R. Sacks, T. Yeung, Exploring graph neural networks for semantic enrichment: Room type classification, Autom. Constr. 134 (2022) 104039, http://dx.doi.org/10.1016/j.autcon.2021.104039, URL https://www.sciencedirect.com/science/article/pii/S0926580521004908.

[34] P. Pauwels, R. de Koning, B. Hendrikx, E. Torta, Live semantic data from building digital twins for robot navigation: Overview of data transfer methods, Adv. Eng. Inform. 56 (2023) 101959, http://dx.doi.org/10.1016/j.aei.2023.101959, URL https://www.sciencedirect.com/science/article/pii/S1474034623000873.

[35] C. Ramonell, R. Chacón, H. Posada, Knowledge graph-based data integration system for digital twins of built assets, Autom. Constr. 156 (2023) 105109, http://dx.doi.org/10.1016/j.autcon.2023.105109, URL https://www.sciencedirect.com/science/article/pii/S0926580523003692.

[36] J. Zhu, N. Nisbet, M. Yin, R. Wei, I. Brilakis, Releasing the power of graph for building information discovery, Autom. Constr. 172 (2025) 106034, http://dx.doi.org/10.1016/j.autcon.2025.106034, URL https://www.sciencedirect.com/science/article/pii/S0926580525000743.

[37] P. Iliadis, E. Bellos, R. Rotas, A. Kitsopoulou, N. Ziozas, N. Nikolopoulos, E. Kosmatopoulos, Comprehensive framework for dynamic energy assessment of building systems using IFC graphs and Modelica, J. Build. Perform. Simul. 18 (4) (2025) 478–499, http://dx.doi.org/10.1080/19401493.2024.2449375.

[38] S. Murali, P. Speciale, M.R. Oswald, M. Pollefeys, Indoor Scan2BIM: Building information models of house interiors, in: 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, 2017, pp. 6126–6133, http://dx.doi.org/10.1109/IROS.2017.8206513.

[39] V. Drobnyi, S. Li, I. Brilakis, Connectivity detection for automatic construction of building geometric digital twins, Autom. Constr. 159 (2024) 105281, http://dx.doi.org/10.1016/j.autcon.2024.105281, URL https://www.sciencedirect.com/science/article/pii/S0926580524000177.

[40] S. Namouchi, I.R. Farah, Graph-based classification and urban modeling of laser scanning and imagery: Toward 3D smart web services, Remote. Sens. 14 (1) (2022) http://dx.doi.org/10.3390/rs14010114, URL https://www.mdpi.com/2072-4292/14/1/114.

[41] S. Jang, G. Lee, J. Oh, J. Lee, B. Koo, Automated detailing of exterior walls using NADIA: Natural-language-based architectural detailing through interaction with AI, Adv. Eng. Inform. 61 (2024) http://dx.doi.org/10.1016/j.aei.2024.102532.

[42] D. Han, W. Zhao, H. Yin, M. Qu, J. Zhu, F. Ma, Y. Ying, A. Pan, Large language models driven BIM-based DfMA method for free-form prefabricated buildings: framework and a usefulness case study, J. Asian Archit. Build. Eng. 24 (3) (2025) 1500–1517, http://dx.doi.org/10.1080/13467581.2024.2329351.

[43] M.Y. Gao, C. Li, F. Petzold, R.L.K. Tiong, Y. Yang, Lifecycle framework for AI-driven parametric generative design in industrialized construction, Autom. Constr. 174 (2025) http://dx.doi.org/10.1016/j.autcon.2025.106146.

[44] J. Zheng, M. Fischer, Dynamic prompt-based virtual assistant framework for BIM information search, Autom. Constr. 155 (2023) 105067, http://dx.doi.org/10.1016/j.autcon.2023.105067, URL https://www.sciencedirect.com/science/article/pii/S0926580523003278.

[45] D. Liu, X. Zhou, Y. Li, An integrated method for BIM data retrieval using large language model, Archit. Sci. Rev. (2025) http://dx.doi.org/10.1080/00038628.2025.2538505.

[46] P. Guo, H. Xue, J. Ma, J.C.P. Cheng, Advancing BIM information retrieval with an LLM-based query-domain-specific language and library code function alignment system, Autom. Constr. 178 (2025) http://dx.doi.org/10.1016/j.autcon.2025.106374.

[47] A. Li, P.K.-Y. Wong, X. Tao, J. Ma, J.C.P. Cheng, An interactive system for 3D spatial relationship query by integrating tree-based element indexing and LLM-based agent, Adv. Eng. Inform. 66 (2025) http://dx.doi.org/10.1016/j.aei.2025.103375.

[48] C. He, W. He, M. Liu, S. Leng, S. Wei, Enriched construction regulation inquiry responses: A hybrid search approach for large language models, J. Manag. Eng. 41 (3) (2025) http://dx.doi.org/10.1061/JMENEA.MEENG-6444.

[49] 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 (7) (2024) http://dx.doi.org/10.3390/buildings14071983.

[50] 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) http://dx.doi.org/10.3390/electronics14112146.

[51] Y. Pan, M. Wang, L. Lu, R. Lamsal, E. Pärn, S. Zlatanova, I. Brilakis, LLM-enabled multi-agent framework for natural language interaction with graph-based digital twins, Autom. Constr. 183 (2026) 106791, http://dx.doi.org/10.1016/j.autcon.2026.106791, URL https://www.sciencedirect.com/science/article/pii/S0926580526000324.

[52] S. Yoon, J. Song, J. Li, Ontology-enabled AI agent-driven intelligent digital twins for building operations and maintenance, J. Build. Eng. 108 (2025) 112802, http://dx.doi.org/10.1016/j.jobe.2025.112802, URL https://www.sciencedirect.com/science/article/pii/S2352710225010393.

[53] S. Bao, H. Bu, Defining and generating operation and maintenance management requirements in digital twin applications using the DT-GPT framework, J. Build. Eng. 104 (2025) 112356, http://dx.doi.org/10.1016/j.jobe.2025.112356, URL https://www.sciencedirect.com/science/article/pii/S2352710225005935.

[54] H. Yang, M. Siew, C. Joe-Wong, An LLM-based digital twin for optimizing human-in-the loop systems, in: 2024 IEEE International Workshop on Foundation Models for Cyber-Physical Systems & Internet of Things, FMSys, 2024, pp. 26–31, http://dx.doi.org/10.1109/FMSys62467.2024.00009.

[55] L. Yang, S. Luo, X. Cheng, L. Yu, Leveraging large language models for enhanced digital twin modeling: Trends, methods, and challenges, 2025, arXiv:2503.02167. URL https://arxiv.org/abs/2503.02167.

[56] S. Gebreab, A. Musamih, K. Salah, R. Jayaraman, D. Boscovic, Accelerating digital twin development with generative AI: A framework for 3D modeling and data integration, IEEE Access 12 (2024) 185918–185936, http://dx.doi.org/10.1109/ACCESS.2024.3514175.

[57] S. Park, C.C. Menassa, V.R. Kamat, Integrating large language models with multimodal virtual reality interfaces to support collaborative human-robot construction work, J. Comput. Civ. Eng. 39 (1) (2025) http://dx.doi.org/10.1061/JCCEE5.CPENG-6106.

[58] Y. Mao, J. Zhong, C. Fang, J. Zheng, R. Tang, H. Zhu, P. Tan, Z. Zhou, SpatialLM: Training large language models for structured indoor modeling, in: Advances in Neural Information Processing Systems, 2025, URL https://arxiv.org/abs/2506.07491.

[59] Q. Team, Qwen2.5: A party of foundation models, 2024, URL https://qwenlm.github.io/blog/qwen2.5/.

[60] Y. Pan, A. Braun, A. Borrmann, I. Brilakis, 3D deep-learning-enhanced void-growing approach in creating geometric digital twins of buildings, Proc. Inst. Civ. Eng.-Smart Infrastruct. Constr. 176 (1) (2023) 24–40, http://dx.doi.org/10.1680/jsmic.21.00035.

[61] I. Armeni, O. Sener, A.R. Zamir, H. Jiang, I. Brilakis, M. Fischer, S. Savarese, 3D semantic parsing of large-scale indoor spaces, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR, 2016, pp. 1534–1543, http://dx.doi.org/10.1109/CVPR.2016.170.

[62] C. Fang, H. Li, Y. Liang, J. Zheng, Y. Mao, Y. Liu, R. Tang, Z. Zhou, P. Tan, SPATIALGEN: Layout-guided 3D indoor scene generation, 2026, arXiv:2509.14981. URL https://arxiv.org/abs/2509.14981.

[63] M. Kellner, M.F. Cervantes, Y. Pan, R. Lu, I. Brilakis, A. Reiterer, SemanticBridge - A dataset for 3D semantic segmentation of bridges and domain gap analysis, 2025, arXiv:2512.15369. URL https://arxiv.org/abs/2512.15369.

[64] M. Kellner, M. Ferrandon Cervantes, Y. Pan, R. Lu, I. Brilakis, A. Reiterer, SemanticBridge—A dataset for 3D semantic segmentation of bridges and domain gap analysis, Dev. Built Environ. 26 (2026) 100912, http://dx.doi.org/10.1016/j.dibe.2026.100912, URL https://www.sciencedirect.com/science/article/pii/S2666165926000700.