UNIVERSITÀ DI TRENTO

DEPARTMENT OF INFORMATION ENGINEERING AND COMPUTER SCIENCE
DEPARTMENT OF INDUSTRIAL ENGINEERING

FONDAZIONE BRUNO KESSLER

Dottorato di Ricerca in Innovazione Industriale / PhD programme in Industrial Innovation

Ciclo 38 / Cycle 38

Scan-to-Energy Digital Twins (Scan-to-EDTs):
Automated Generation of Energy Digital Twins from 3D point clouds

Oscar Roman

Supervisor/Tutor: Fabio Remondino

Co-supervisor/Co-tutor: Elisa Mariarosaria Farella

Coordinatore / Coordinator: Prof. Domenico Siracusa

ANNO ACCADEMICO / ACADEMIC YEAR
2024/2025

Finanziato dall'Unione europea NextGenerationEU | Ministero dell'Università e della Ricerca | Italiadomani | UNIVERSITÀ DI TRENTO

To my family, whose love, sacrifices and faith
in me have been my greatest strength
throughout this journey.

Abstract

Digital Twins (DTs) are transforming construction and energy management sectors by integrating 3D surveying, monitoring, Building Performance Simulation (BPS) and Building Energy Simulation (BES) from the earliest design or retrofit stages. Moreover, dynamic thermal simulations further support energy performance assessments by modeling indoor conditions to meet comfort and efficiency targets. Yet, their reliability depends on accurate, standards-compliant building 3D models, which are costly to be created.

This research introduces a complete framework for automatically generating energy-focused Digital Twins (EDTs) directly from unstructured point clouds. Combining Deep Learning–based instance detection, Scan-to-BIM techniques and computational geometry, the method produces simulation-ready models without manual intervention. The resulting EDTs streamline early-stage performance evaluation, enable scenario testing and enhance decision-making for energy-efficient retrofits, advancing smart-building design through predictive simulation.

Keywords

Automation in constructions, BIM, BEM, Deep Learning, 3D reconstruction, Multi-level Digital Twins, Energy simulation, IoT, Open source.

TABLE OF CONTENTS

ABBREVIATIONS & LIST OF TERMINOLOGY

Model Types for Built Environment Analysis and Simulation

Building Information Modelling (BIM):
A shared, structured model that stores geometry, properties, relationships and timelines for a facility across design, construction and operations. Used for clash checks, quantities, schedules and asset data.

Building Energy Model (BEM):
A physics-based model of a building used to simulate loads, HVAC behavior, controls and energy use over time. Outputs include hourly energy, peak demand, comfort metrics.

Mechanical, Electrical, Plumbing (MEP):
The mechanical, electrical and plumbing scopes modeled and coordinated inside BIM. Focus is on routing, clearances, clashes and performance requirements.

Federated model: Multiple discipline models linked by shared coordinates into one view without merging files. Each team maintains its own file, which preserves authorship and traceability.

Finite Element Model / Finite Element Analysis (FEM / FEA)
A numerical modeling approach that discretizes a continuous domain into elements and nodes to approximate the solution of partial differential equations governing physical phenomena. Accuracy depends on mesh resolution, element formulation, material properties, and boundary conditions.

Boundary Representation (B-Rep)
A geometric modeling scheme in which solids are represented by their enclosing surfaces, edges, and vertices. B-Rep models explicitly encode topology and are commonly used in CAD, BIM, and simulation workflows.

Constructive Solid Geometry (CSG)
A solid modeling technique that represents complex geometries such as Boolean combinations of primitive solids (e.g., union, intersection, difference). CSG emphasizes parametric definition over explicit surface topology.

Graph-Based Model
A data representation in which building elements are modeled as nodes and their geometric, topological, and semantic relationships are encoded as edges. Graph-based models enable efficient querying, validation, traversal, and integration across multiple abstraction levels and are well suited for Digital Twin workflows.

Simulation-Ready Model
A building model that satisfies the geometric, topological, and semantic requirements of a target simulation engine, including watertight geometry, valid adjacencies, consistent zoning, and complete material and boundary condition definitions.

Digital Twin (DT)
A dynamic digital representation of a physical asset that integrates geometry, semantics, state, and time, and is continuously updated through data exchange with the physical system. DTs support monitoring, simulation, prediction, and decision-making across the asset lifecycle.

Format files

IFC (Industry Foundation Classes)
An open, standardized data model for describing building and construction information, designed to support interoperability across BIM platforms. IFC encodes geometry, semantics, relationships, and processes using an object-based schema.

gbXML (Green Building XML)
An open XML-based schema designed to transfer building geometry and space definitions from BIM tools to energy and environmental analysis software. gbXML emphasizes zones, surfaces, and adjacencies but relies on simplified planar geometry.

epJSON (EnergyPlus JSON)
A JSON-based input format for the EnergyPlus simulation engine, used to define building energy models, systems, schedules, and simulation parameters in a structured, machine-readable form.

WKT (Well-Known Text)
A text-based representation of vector geometry objects such as points, lines, and polygons, commonly used in GIS and spatial databases.

XML (Extensible Markup Language)
A hierarchical, tag-based markup language used to encode structured data with explicit schemas and validation rules.

Types of energy analysis

Monitoring
The continuous or periodic acquisition and analysis of operational building data, such as temperature, humidity, energy consumption, and system states, typically sourced from sensors and building management systems.

Simulation (Static and Dynamic)
- Static Simulation: Analysis performed at a fixed point in time or under steady-state assumptions.
- Dynamic Simulation: Time-dependent simulation capturing transient behavior, system dynamics, and temporal interactions.

Predictive Maintenance
A maintenance strategy that leverages sensor data, historical trends, and analytical models to anticipate equipment failures and schedule interventions proactively.

Computational Fluid Dynamic (CFD)
The numerical simulation and analysis of airflow, heat transfer, and fluid-related phenomena within and around buildings, used to assess ventilation performance, thermal comfort, indoor air quality, and localized heat exchange under steady-state or transient conditions.

Model structure and properties

Topology
The explicit representation of adjacency and incidence relationships between vertices, edges, faces, and volumes. Topology enables robust spatial reasoning, validation of connectivity, and identification of internal and external boundaries.

Hierarchy
A structured containment model describing nested spatial relationships (e.g., *site → building → storey → space → surface*). Hierarchy supports aggregation, inheritance of properties, and scalable querying across model levels.

Manifold Geometry
A geometric condition in which each edge is shared by exactly two faces, ensuring a well-defined surface suitable for simulation and analysis.

Watertight Geometry
A property of a 3D model in which all surfaces form a closed, gap-free shell that fully encloses a volume, required for volumetric simulation and physical analysis.

Thematic Surfaces
Surfaces labeled according to their functional or semantic role (e.g., wall, roof, floor), enabling classification, semantic reasoning, and domain-specific analysis.

Topologically Compliant Model
A geometric model that explicitly encodes adjacency, incidence, and containment relationships and satisfies manifold and watertight conditions. Such models guarantee consistent connectivity between spaces and surfaces, enabling reliable physics-based simulation, semantic querying, and Digital Twin integration.

Core Algorithms and Geometric Methods

RANSAC: Robust estimator that repeatedly samples minimal sets, fits a model, counts inliers and keeps the model with the largest consensus set.

Principal Component Analysis (PCA): Eigen decomposition of the covariance matrix to find directions of maximum variance, used for dimension reduction and feature decorrelation.

Region Growing Segmentation
Groups points based on proximity and similarity (normal vectors, curvature). Common for walls, floors, ceilings.

Normal Estimation
Computes local surface orientation using k-nearest neighbors or PCA. Fundamental for plane detection and classification.

Plane Extraction / Planar Segmentation
Often built on RANSAC; explicitly mention as a method since planes are central to walls, floors, slabs.

Graph Traversal Algorithms (BFS / DFS)
Used for connectivity checks, room adjacency, zone traversal, and validation.

Connected Components Analysis
Identifies separate rooms, spaces, or building parts from graphs or meshes.

Cycle Detection in Graphs
Important for validating closed floorplan loops and topological consistency.

Voxelization / Octree Decomposition

Discretizes space for acceleration, noise filtering, occupancy reasoning, and scale management.

Gaussian density distribution: Models point intensity with a Gaussian or a mixture of Gaussians. Used for clustering, outlier detection and occupancy estimation.

Edge walking: Traverses connected edges in polygons or meshes to follow boundaries, extract contours, or build loops.

Double Connected Edge List (DCEL): Half edge structure with explicit twin, next and previous pointers, providing fast navigation among faces, edges and vertices in planar subdivisions.

Ray tracing: Casts rays and computes intersections with geometry. Used for rendering, visibility checks, daylight studies and collision queries.

Convex hull: The minimal convex set that contains all points. Useful for bounds, collision broad phase and shape descriptors.

1

This chapter frames the challenge and rationale for the PhD. Buildings are central to the EU's net-zero 2050 agenda, yet a largely older building stock and fragmented digital workflows slow progress on efficiency upgrades. The chapter reviews the policy context, the role of Building Information Models (BIM) in the Digital Twins (DTs) workflows and the limitations of current toolchains that trap data in siloed, visualization-only platforms. Finally, the chapter outlines the thesis contribution: an end-to-end framework that turns reality-captured point clouds and BIM into auditable energy models and actionable, portfolio-scale retrofit decisions, paving the way toward a digital building logbook (DBL).

Chapter 1

1. Introduction

1.1 Background and motivation

Climate change is one of the most urgent challenges of our time, driven by population growth and the continuous expansion of building stock. As urbanization increases, buildings have become a major source of environmental pressure at local, regional, and global scales, largely due to high energy demand and continued reliance on fossil fuels, particularly natural gas and oil.
As shown in Fig. 1, building energy use is dominated by heating and cooling, while electricity demand continues to rise. Significant regional differences in primary energy consumption, with consistently high levels in Asia Pacific and Europe, further highlight the global scale of the challenge. In response, governments worldwide have promoted, through policy initiatives and regulatory mandates, the adoption of advanced solutions to reduce fossil fuel dependency, improve energy efficiency, and enhance overall urban performance[1].

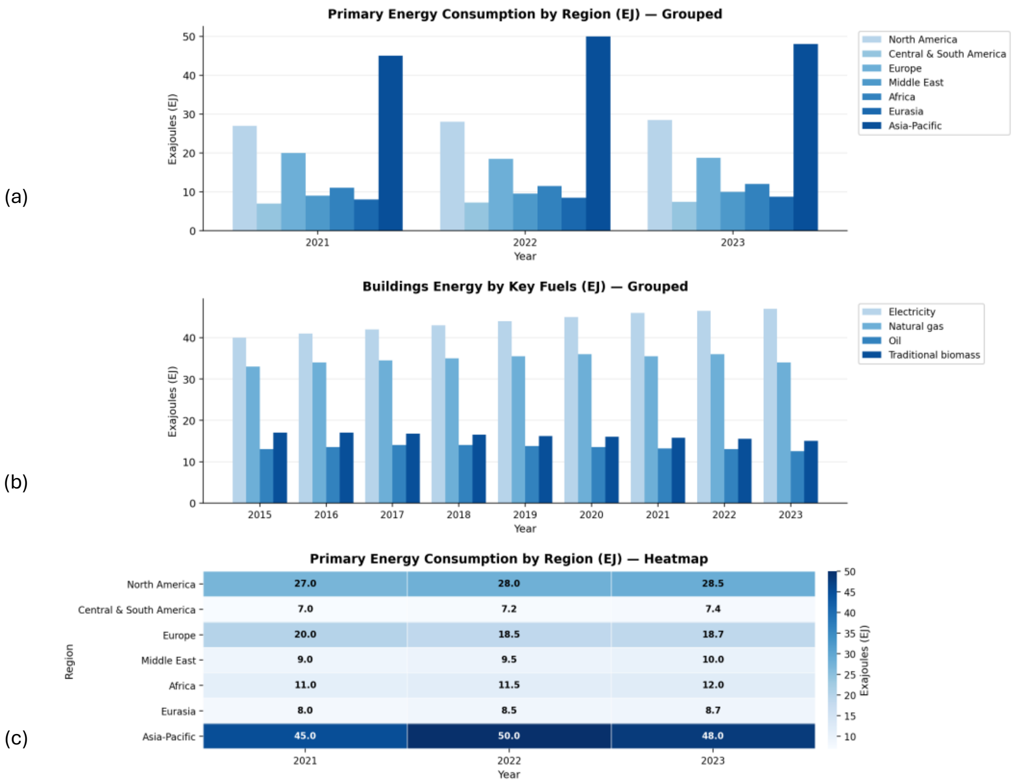


Fig. 1. Energy consumption trends by region and fuel. **(a)** Grouped bars of primary energy consumption by region (2021–2023). **(b)** Buildings energy use by key fuels (2015–2023). **(c)** Heatmap of regional primary energy consumption, highlighting relative magnitudes and temporal variation.

In Europe, the building sector accounts for approximately 30% of final energy consumption and 26% of energy-related greenhouse gas emissions, making it a central focus of climate mitigation efforts.
Buildings are also responsible for around 40% of total energy consumption and 50% of gas use, while nearly 80% of household energy demand is associated with heating, cooling, and domestic hot water.
Despite this significant impact, the annual energy renovation rate remains very low at about 1%[2].
To meet the EU's 2050 climate neutrality targets, this rate must increase to approximately 3% annually, a threefold acceleration that requires not only policy support but also technical tools enabling rapid, reliable

[1] https://www.iea.org/energy-system/buildings
[2] https://energy.ec.europa.eu/topics/energy-efficiency/energy-performance-buildings/energy-performance-buildings-directive_en

assessment of renovation options at building and portfolio scales. Reaching the European Union's 2050 climate neutrality target, therefore, depends on the large-scale renovation of the existing building stock and the adoption of nearly zero energy buildings (Fig. 2). As most buildings were constructed before 2000 and many exhibit poor energy performance, the revised Energy Performance of Buildings Directive (EPBD) prioritizes deep renovation, improved performance assessment, and the integration of smart and renewable-ready building systems.

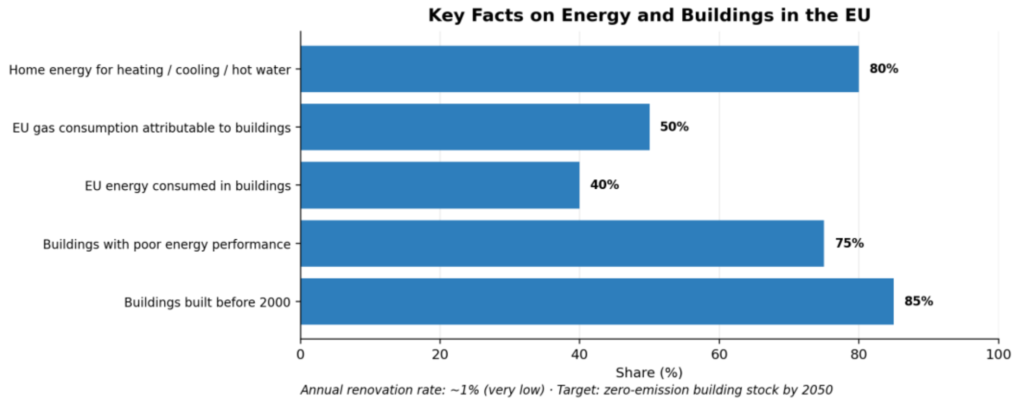


Fig. 2. Key energy and building performance indicators in the EU.

In particular, the revised EPBD[3], which entered into force on 28 May 2024 and must be transposed into national law by 29 May 2026, represents a key step in accelerating building renovation across the European Union. It prioritizes the renovation of the worst-performing buildings while allowing Member States to tailor measures to national building stocks, climatic conditions, and socioeconomic contexts (Table 1).
Under the directive, each Member State is required to develop a national building renovation plan defining its strategy to decarbonize the building stock, including financing mechanisms and workforce development.

Building type	Target year	Requirement
Non-residential	2030	Renovation of the 16% worst-performing buildings
Non-residential	2033	Renovation of the 26% worst-performing buildings
Residential	2030	16% reduction in average primary energy use
Residential	2035	20–22% reduction in average primary energy use
Residential (scope)	-	≥55% of savings from renovation of worst-performing housing stocks

Table 1. European Union targets for renovation rates and primary energy reduction in residential and non-residential buildings.

To support implementation, the directive strengthens energy performance certificates using a common EU framework and introduces building renovation passports to guide staged renovations. Exemptions are permitted for specific building categories, such as heritage buildings, ensuring flexibility while maintaining ambitious decarbonization goals.

In this context, energy simulation prior to intervention plays a key role.
Retrofit decisions involve substantial investment and long-term consequences. and accurate simulations support the evaluation of retrofit options, cost-benefit optimization and verification of expected energy savings before implementation.
However, current energy simulation workflows rely on high-quality, well-structured geometric and semantic data that are rarely available for existing buildings, creating a major barrier to scalable, reliable and informed retrofit planning.

[3] https://eur-lex.europa.eu/legal-content/EN/TXT/

To overcome these limitations and support lifecycle-oriented building intelligence, the European Union is promoting the Digital Building Logbook (DBL)[4] as a centralized digital backbone integrating existing building records, digital acquisition technologies, and sustainability frameworks.
Despite this vision, practical implementation remains challenging due to fragmented workflows and unreliable data exchange between BIM simulation and operational systems, often leading to semantic loss and manual rework.
This work directly addresses these gaps by enabling robust, interoperable and scalable integration of geometry, semantics, and simulation data, supporting reliable energy assessment and renovation planning at building and portfolio scales.

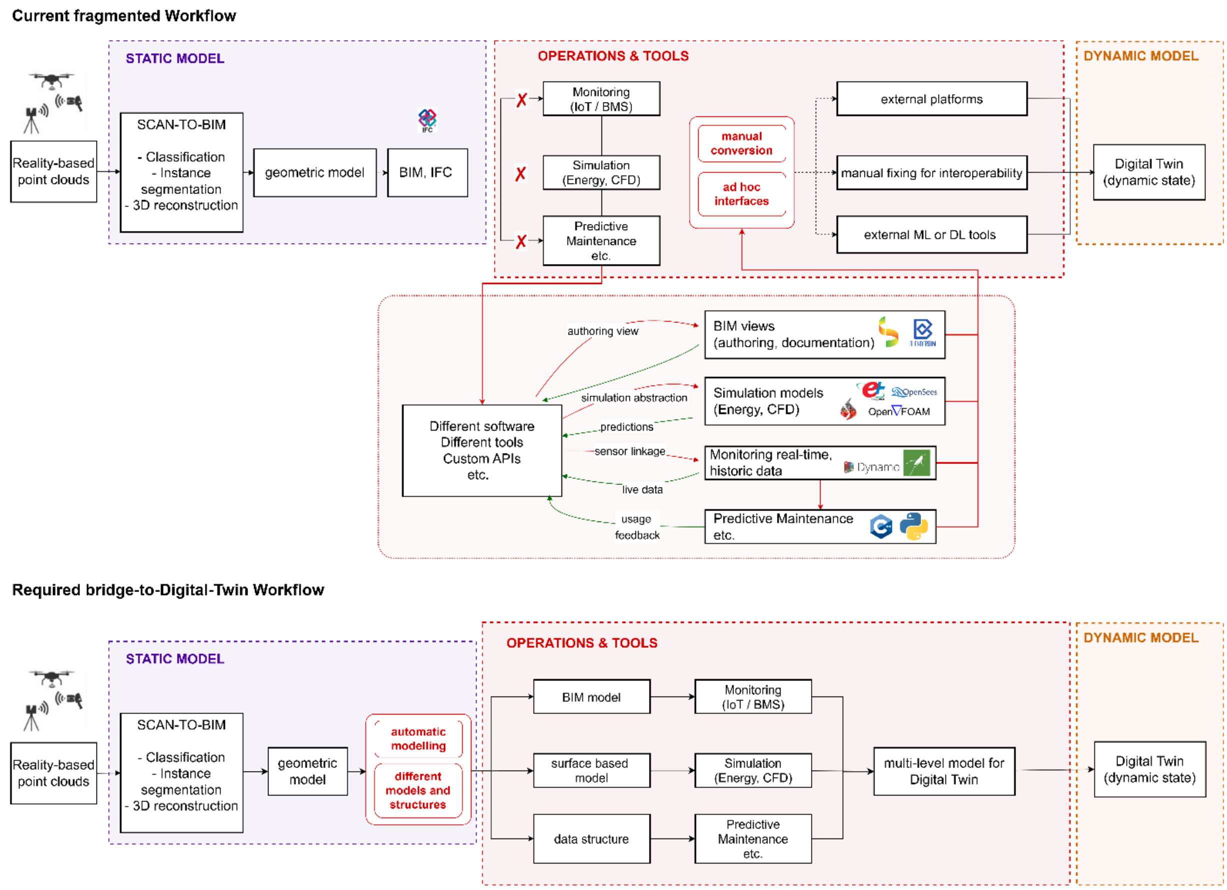


Fig. 3. Comparison of (a) current fragmented practice and (b) an idealized reference workflow toward Digital Twin development. Unlike conventional workflows based on disconnected tools and manual transformations, the proposed approach relies on a single workflow for the generation of energy-oriented Digital Twin models.

1.2 Context

Within the Architecture, Engineering, Construction, Owners and Operations (AECOO) sector, digital tools are developed for distinct purposes and therefore rely on fundamentally different and often incompatible data models (Fig. 3).

- *Building Information Modeling (BIM)* platforms are optimized for design authoring, coordination, and documentation, relying on proprietary parametric solid representations and hierarchical spatial structures.
- *Energy modeling tools*, in contrast, operate on simplified boundary representations that enforce strict topological constraints, requiring watertight geometries, explicit adjacencies, thermal zoning, and validated construction assemblies.

[4] https://data.europa.eu/doi/10.2826/519144

- *Digital Twin (DT)* systems introduce an additional set of requirements, centered on dynamic synchronization with sensor data, temporal versioning, and bidirectional coupling between physical operation and digital state.

These differences reflect divergent priorities: BIM emphasizes constructability and documentation, energy modeling prioritizes physical accuracy and simulation fidelity, and DT systems focus on real-time responsiveness and operational intelligence. Real-world conditions further exacerbate these challenges, as existing buildings often deviate from design intent, documentation is incomplete or outdated, and operational data are fragmented across heterogeneous systems and vendors, often constrained by ownership and privacy concerns.
As a result, no single tool or data format can natively support the full DT lifecycle. In practice, users rely on complex, multi-step conversion workflows that introduce systematic errors and require extensive manual intervention. Models derived from reality-capture data are repeatedly exported into different formats to satisfy heterogeneous software requirements, leading to duplicated representations, loss of structural consistency, and progressive divergence between the original model and downstream derivatives (Fig 3).
The following section examines these barriers in detail through three interconnected problems that collectively prevent reliable DT implementation.

1.3 Problem Statement

The fragmentation described in Section 1.2 manifests as three interconnected barriers that collectively prevent the reliable and scalable implementation of building DTs. These problems operate at different levels, such as conceptual, technical, and methodological, but reinforce one another, creating a systemic impasse in current practice (Fig. 4).

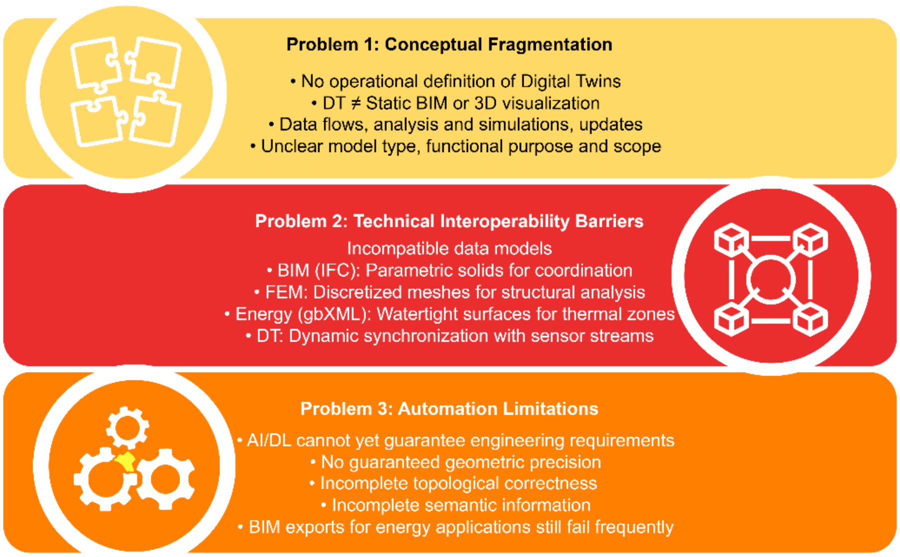


Fig. 4. Core problems identified in building DT development, highlighting conceptual fragmentation, technical interoperability barriers, and automation limitations.

Problem 1: Conceptual Fragmentation

No operational definition exists for building DTs.
The term *"Digital Twin"* is used inconsistently across literature and practice, often confused or equated with static BIM models or 3D visualizations.
Missing are concrete specifications of core components, required data flows, update mechanisms and functional capabilities that distinguish a DT from conventional 3D digital models. Without this clarity, standardization remains impossible, and practitioners cannot reliably scope, implement, validate or compare different DT systems.

Furthermore, the functional purpose of DTs remains ambiguous. For engineering applications, a DT must focus on specific objectives such as energy optimization, predictive maintenance, or operational monitoring. Yet current approaches lack frameworks for determining which abstractions, data links, and temporal resolutions are required for meaningful building applications.

Problem 2: Technical Interoperability Barriers

BIM platforms, finite element models, energy simulation tools, and DT applications rely on data models that are fundamentally incompatible in both structure and intent.
Each domain optimizes its representations for distinct objectives, which prevents seamless data reuse:

- *BIM platforms* use proprietary parametric solids representations optimized for design coordination, quantity extraction and construction documentation.
- *Finite element models (FEM)* rely on discretized representations based on nodes and elements, mid-surface abstractions, and physics-driven connectivity, specifically designed for structural and physical analysis.
- *Energy modeling tools* require watertight boundary surfaces, based on topology, with explicit thermal zones, validated adjacencies, and physics-based connectivity.
- *Digital Twins* demand dynamic synchronization with sensor streams, versioning across time, and bidirectional coupling between physical operations and digital state.

	BIM - IFC	**FEM**	**gbXML**	**epJSON**
Format file	STEP (EXPRESS)	INP (ABAQUS input), MSH (mesh)	XML (XSD schema based)	JSON
Encoding	Plain text, line-based	Plain text blocks (cards, keywords, lists)	XML tags	JSON with nested keys
Structure	Object graph network	Nodal–element connectivity matrix	Hierarchical Tree	Key-value
Application	BIM coordination	Structural or physical simulation	Energy modelling	Energy simulation
Orientation	Geometry-oriented	Analysis and physics oriented	Space and zone centric	Simulation oriented
Geometry detail	High/medium detail (LOD)	Simplified discretized meshes (nodes and elements)	Planar simplified surfaces	Planar geometry, vertex array

Table 2. Comparison of data exchange across BIM, structural analysis, and energy simulation domains, showing fundamental incompatibilities in encoding, structure and geometric representation that prevent seamless model exchange.

Table 2 further demonstrates that IFC, FEM formats, gbXML and epJSON differ fundamentally in file encoding, internal structure, application focus, orientation, and required geometric detail. IFC prioritizes object-rich geometry for coordination, FEM formats emphasize discretized meshes for physical analysis, gbXML adopts a space and zone-centric abstraction for energy modeling, and epJSON is optimized for simulation execution rather than geometric fidelity.
The radar chart (Fig. 5) compares IFC, FEM, gbXML, and epJSON in terms of geometric detail, topology, semantics and energy orientation, revealing their fundamentally different design priorities. The conceptual positioning diagram further shows a progression from geometry-oriented BIM representations toward task-specific, simulation-driven formats, underscoring why seamless model exchange remains challenging.
These structural incompatibilities create systematic data loss during model exchange. Exports from BIM to energy simulation formats (IFC → gbXML → EnergyPlus) routinely produce missing geometry, broken topology, corrupted adjacencies, and invalid thermal zones (Fig. 6).

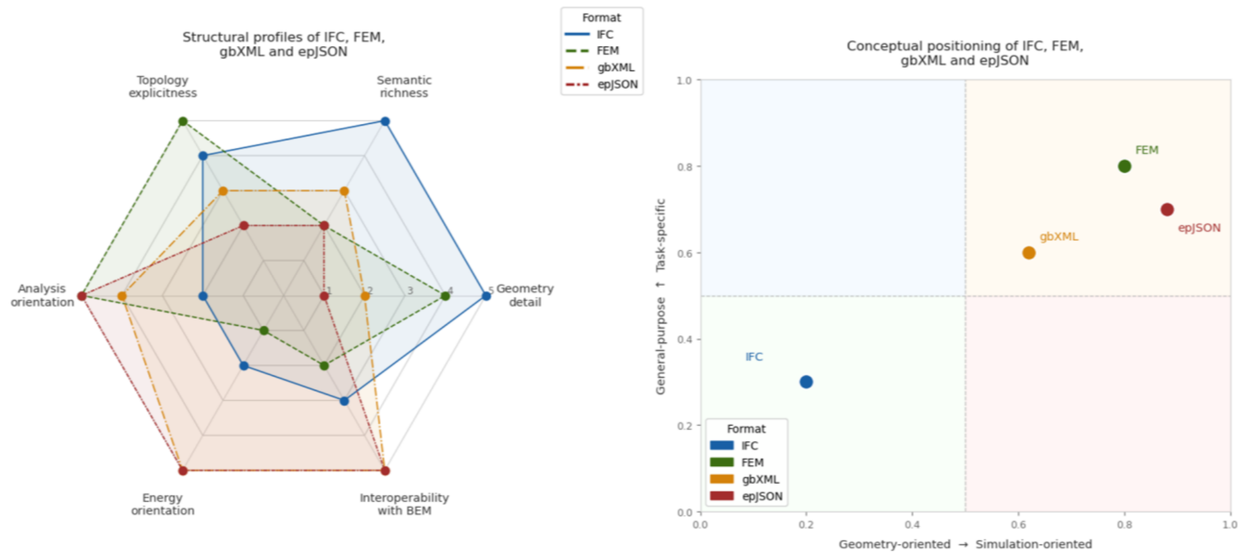


Fig. 5. Comparison of IFC, FEM, gbXML and epJSON in terms of representation logic, intended application domain and level of geometric abstraction. The figure highlights why these formats should be treated as task-specific outputs rather than as interchangeable core representations.

Problem 3: Limitations of automation in construction

Methodological and automation limitations further constrain scalability.
While DL-based approaches have advanced classification and 3D reconstruction from point clouds, they remain unable to guarantee the topological correctness, geometric precision, and semantic completeness required for physics-based simulation and DT integration. Even manual BIM exports frequently yield inconsistent models, reinforcing dependence on authoring tools ill-suited for existing buildings.

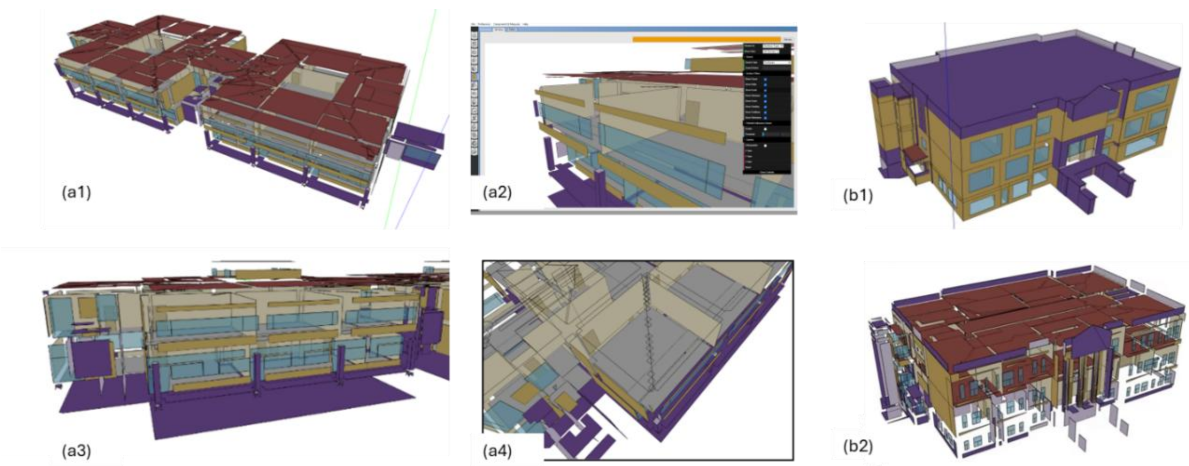


Fig. 6. Examples of geometric and topological inconsistencies in automatically exported gbXML models from BIM software: (a1–a4) typical errors in exporting geometries from BIM software affecting energy analysis; (b1) the correct Ground Truth model; (b2) automatically exported gbXML result, inconsistent geometries.

These three problems, (i) conceptual ambiguity around what DTs are, (ii) technical incompatibilities preventing reliable data exchange, and (iii) automation limitations in AI-based reconstruction, together define the research gap this thesis addresses. The following objectives and research questions are structured to systematically resolve each barrier while demonstrating their integration in a unified framework.

1.4 Research Questions and Objectives

This thesis develops an end-to-end framework that converts reality-captured data (point clouds) into simulation-ready, DT-enabled building models.
The framework addresses the three identified problems through five integrated research objectives:

Objective I: Define Energy Digital Twins

RQ1: What essential components, data flows and modeling methods formalize an operational definition of an Energy Digital Twin (EDT) and consistently distinguish it from static BIM or 3D models?

Objective II: Clarify BIM's Role in DT Development

RQ2: Under which conditions is BIM necessary for DT creation, and when can alternative models (mesh-based, graph-based, city-scale) serve as foundations?
RQ3: How do BIM-derived representations compare with alternatives in geometric accuracy, semantic richness and suitability for integrating real-time data streams in DT environments?

Objective III: Establish Automation and AI Boundaries

RQ4: What can DL reliably extract from point clouds and images, and where does it fail in terms of topology, accuracy, and semantic consistency for engineering applications?
RQ5: How can DL outputs be integrated with deterministic methods to produce simulation-ready, DT-ready models with guaranteed geometric validity?

Objective IV: Define DT Functional Scope

RQ6: What core functions should build DTs fulfill, and which abstractions of geometry, semantics, and system behavior balance model complexity with practical usability?
RQ7: What mechanisms ensure DTs remain accurate, evolving proxies of physical assets rather than static snapshots?

Objective V: Ensure Robust Interoperability

RQ8: Which modeling and exchange strategies minimize data loss and preserve geometric/semantic integrity across BIM, simulation, and DT workflows?
RQ9: How can automated validation, correction, and synchronization prevent repeated remodeling and maintain reliable, reproducible dataflows?

The analysis and discussion of each research question articulated in this study are systematically presented in Section 7.2.

1.5 Proposed Framework

To address all three identified problem areas simultaneously, this thesis proposes an integrated Scan-to-EDT pipeline that unifies previously fragmented workflows into a single, automated processing chain. Rather than treating BIM generation, energy modeling, and DT creation as separate processes requiring manual bridges, the framework maintains structural consistency across all representations through a graph-based data backbone.
This approach transforms raw reality-captured 3D data into operational DT models suitable for energy analysis, simulation, and lifecycle management.

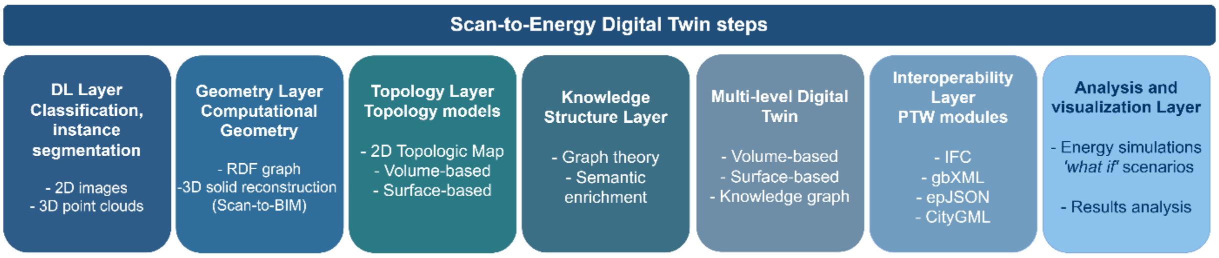


Fig. 7. Scan-to-EDTs framework integrating reconstruction, topology, semantics and interoperability into a unified, continuous workflow.

The pipeline comprises eight interconnected stages (Figure 7):

1. Point cloud classification and panoptic segmentation automatically identify structural elements (walls, floors, ceilings, openings) and building objects from laser scanning or photogrammetry data, addressing the automation challenge (Problem 3).
2. Device mapping annotation uses domain-informed APIs to manually detect and annotate indoor devices (HVAC terminals, lighting, sensors), incorporating expert knowledge where DL remains unreliable.
3. Geometric reconstruction leverages computational methods to reconstruct geometries on a per-class basis, using specialized reconstruction modules designed to address the specific characteristics of each class (https://github.com/Saiga1105/Scan-to-BIM-CVPR-2024).
4. Topological modeling generates both volumetric models (rooms, thermal zones) and surface-based thematic representations (walls, slabs, roofs) with explicit adjacencies and validated connectivity with guaranteed topological properties, producing watertight, manifold representations suitable for physics-based simulation (Problem 2).
5. Graph-based integration links geometries, devices, metadata and relationships in a unified semantic structure, providing the flexible, extensible data backbone required for DT operations and clarifying the DT's structural definition (Problem 1).
6. Semantic enrichment attaches materials, thermal properties, construction assemblies, and operational attributes through the graph structure, enabling energy simulation without manual remodeling.
7. Scenario generation supports *"what-if"* energy analysis by allowing parametric variation of the envelope properties, system configurations and operational schedules, directly addressing the need for simulation before intervention.
8. Multi-format export via Parser-Transform-Writer modules translates the graph-based DT into IFC (for BIM interoperability), gbXML and epJSON (for energy simulation) and CityGML (for urban-scale integration), ensuring compatibility across the full AECOO toolchain without data loss.

The framework produces simulation-ready models that serve as operational DT backbones, functioning as dynamic repositories for both geometric and semantic data and real-time operational streams.
By integrating reality-captured point clouds, semantic modeling, and energy simulation in a single pipeline (Fig. 8), it provides a concrete implementation pathway for the DBL vision while establishing clear answers to the conceptual, technical, and automation challenges identified in Section 1.2.

1.6 Thesis Structure

This thesis is organized into seven chapters that progress from conceptual foundations through methodology to implementation and validation.
After the introduction (Chapter 1), the thesis is organized as follows:

- **Chapter 2:** Fundamentals on Building DTs establish the theoretical basis for DT creation followed on the presented work, covering (i) semantic segmentation and classification methods, (ii) computational geometry for 3D reconstruction, (iii) BIM-based and BEM-based modeling approaches, (iv) DT typologies for different purposes, (v) Topologic-based representations, (vi) information-enrichment strategies and (vii) representative DT applications in the building sector.

- **Chapter 3:** Challenges and Barriers in Current DT Workflows analyzes the technical, methodological, and operational limitations affecting existing DT pipelines, examining fragmentation across tools, interoperability failures, systematic data loss, computational constraints, DL limitations in engineering-based models for applications and simulations, and the absence of integrated end-to-end frameworks.

- **Chapter 4:** Methodology presents the complete Scan-to-EDT framework in detail, describing the classification pipeline, computational geometry reconstruction algorithms, Topologic map generation, multilevel DT model structures, ILD-based semantic enrichment procedures and the graph-based DT architecture that enables interoperability and scenario analysis.

- **Chapter 5:** Results reports experimental outcomes for each pipeline stage: classification accuracy, geometric reconstruction quality, Topologic modeling validation and solid model generation. Furthermore, it demonstrates the Parser-Transform-Writer modules for exporting to IFC, gbXML, epJSON and CityGML formats, and presents the resulting operational DT with integrated energy simulation capabilities.

- **Chapter 6:** Advancements, Limitations and Main Contributions synthesize the methodological innovations introduced by the research, critically examines limitations and challenges encountered during implementation, and discusses their implications for future DT research and professional practice.

- **Chapter 7:** Conclusions provide a comprehensive synthesis of findings, directly answers each research question posed in this introduction, evaluates the framework's contribution to closing the identified gaps, and outlines promising directions for future research in building DTs development and deployment.

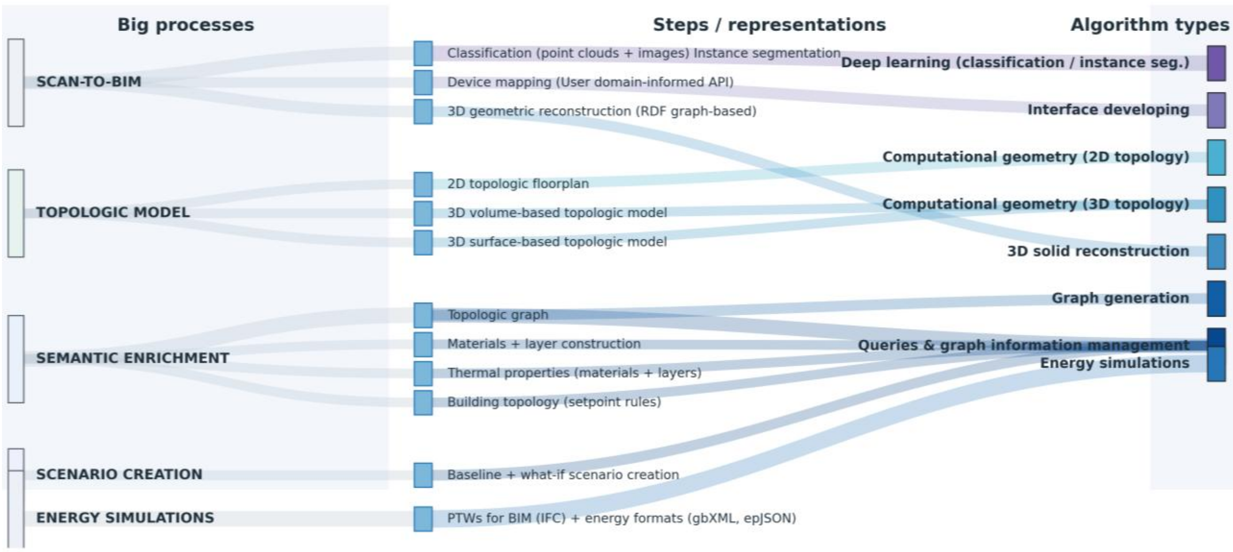


Fig. 8. Overview of the Scan-to-EDTs workflow, linking major process stages to their corresponding representations and algorithmic methods.

2

Chapter 2 introduces the methodological and technological foundations of the Scan-to-Energy Digital Twins (Scan-to-EDTs) framework. The chapter reviews the state of the art across the main workflow stages required for energy-oriented Digital Twin generation, from data acquisition and geometry reconstruction to topology modeling, semantic enrichment, and simulation-driven analysis. Emphasis is placed on the role of multi-source sensing, computational geometry, and knowledge-based representations in enabling interoperable and energy-compliant Digital Twins. By structuring the review according to progressive model abstractions, the chapter clarifies how different reconstruction and modeling strategies contribute to downstream energy performance assessment. This systematic overview establishes the conceptual and technical background for the proposed framework and supports the integration of heterogeneous methods into a coherent Scan-to-EDTs pipeline.

Chapter 2

2. Fundamentals on Digital Twin creation

The Digital Twin (DT) concept gained recognition in 2002 following a presentation by Michael Grieves at the University of Michigan (Grieves, 2016). Its conceptual origins can be traced to NASA's Apollo 13 mission, which employed paired physical and virtual systems for real-time monitoring (Allen, 2021).
A DT is a digital replica of a physical asset (Sharma et al., 2022; Arsecularatne et al., 2024; Wysocki et al., 2025) that represents its geometry, physical and operational properties, and contextual relationships.
As demonstrated in Chapter 1 through the identified problems, and due to this inherent complexity, DTs cannot be implemented as single models. Instead, as illustrated in Figure 7, they require structured and multi-level frameworks with clearly defined and integrated development stages.
However, such hierarchical and integrated approaches remain uncommon in current DT workflows, which often address isolated components rather than coherent systems (Tuhaise et al., 2023). This lack of systematic integration limits internal consistency and the ability to represent asset behavior across scales and use cases.
Crucially, DTs must extend beyond visually enriched three-dimensional models to support monitoring, simulation, and at advanced maturity levels, prediction and optimization through collaborative visualization. Traditional tools such as BIM, which primarily function as static information repositories, are therefore insufficient on their own and must be embedded within broader DT platforms that integrate modeling, data management, analytics, and, most importantly, simulation.

2.1 Digital Twin types and main processes involved

Typical main processes involved in DT generation workflows (Klar et al., 2024; Wang et al., 2024) can be summarized as in Figure 9. These workflows exhibit variations in scope and implementation depending on project requirements, available resources, and intended DT maturity level (Section 2.3), but in general they adopt BIM models as a central component, which serve as the basis for generating alternative representations required for analysis, simulation, or visualization.

Fig. 9. The conventional workflow for DT development, centered on BIM as the primary model backbone.

These workflows typically start from available geometric data, which may include high-resolution point clouds acquired through 3D reality-based techniques such as laser scanning or photogrammetry. When present, point clouds provide a dense geometric basis for DT generation. However, in many cases the workflow proceeds without direct reality-captured input, relying instead on existing design or as-built documentation. Then the workflows involve:

1. *Building Information Model (BIM):* The point cloud is converted into a BIM model, often manually or through semi-automated processes, where architectural elements, spaces, materials and properties are organized in a structured, semantic building model with solid geometries.

2. *IoT Sensor Integration:* Data from sensors monitoring temperature, humidity, occupancy and energy consumption are collected in a cloud environment. These data streams are then, typically through external platforms or custom visualization dashboards, associated with the corresponding BIM elements, thereby establishing a quasi-real-time data link between the physical building and its digital representation. In current practice, however, this integration is very often only partial, limited in terms of sensor coverage, data types, interoperability or temporal continuity.
3. *Digital Twin Generation:* Geometry, semantics, and sensor data, recreated from scratch or derived from IFC/BIM, are integrated into a dynamic model within external tools or customized APIs, in which physical and digital states are synchronized, enabling continuous updating of the DT.
4. *Platforms and Data Visualization:* The DT is deployed on platforms such as cloud services, web dashboards and 3D visualization tools, allowing users to access, explore, and manage building data and model behaviour.
5. *Digital Twin Applications:* On top of this infrastructure, applications are implemented for real-time monitoring, simulations, and predictive maintenance, turning the DT into a decision-support and optimization tool.

To simplify the integration of time-varying data with geometric models, some workflows introduce the concept of a Digital Shadow (DS) (Liebenberg & Jarke, 2023). A DS represents the continuous stream of time-stamped measurements and events generated by the physical asset. It does not store geometry itself but references geometric and semantic elements of the DT through stable identifiers. In practice, including in the present work (Chapter 4), the DS is often integrated into the DT model by binding its time-varying data streams to specific geometric and semantic entities.
When such an integrated model supports data analysis, simulation and prediction, it evolves into a fully operational DT (Fig. 10), characterized by bidirectional feedback loops for state estimation, simulation, and control. Advanced data-driven methods, including Artificial Intelligence (AI), Machine Learning (ML), Deep Learning (DL) and Physics-Informed Neural Networks (PINNs), further DT capabilities by enabling real-time monitoring (Hu et al., 2024), predictive maintenance (Cummins et al., 2024), energy optimization (Liu et al., 2025), and adaptive system behavior. In particular, PINN-driven data-driven decision support systems (DDSS) facilitate continuous learning and physics-consistent optimization, thereby significantly improving the robustness and autonomy of DT-enabled systems (Pavirani et al., 2024).

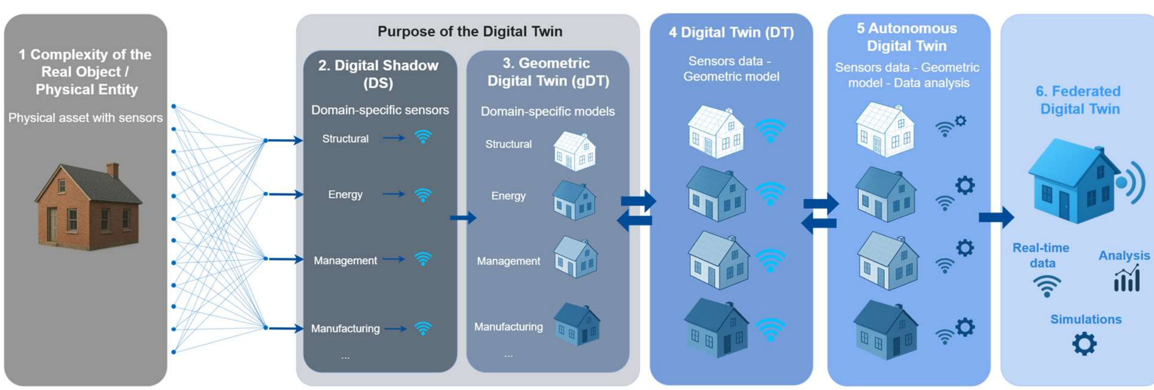


Fig. 10. Main workflow schema for DT to bridge physical and digital assets.

At larger scales, multiple DT instances can be interconnected to form Federated Digital Twins (FDTs), linking domain-specific twins (e.g., building, structural, or system-level DTs) or aggregating individual building DTs at district and city scales. This federation enables cross-domain and cross-scale analysis, supporting coordinated monitoring, predictive maintenance, and energy optimization. Integration with Building Management Systems (BMS) allows real-time operation, while PINN-based data-driven decision support systems improve robustness by embedding physical constraints within learning-based models.

2.2 Digital Twin applications in the built environment

DTs are becoming one of the most widely used models in the built environment for structural analysis (Hu et al., 2024), monitoring (Lu et al., 2020; Kreuzer, et al., 2024; Armijo & Zamora-Sánchez, 2024), simulations (Naeem et al., 2025), and predictive maintenance (Attaran et al., 2024; Meddaoui et al., 2024; Asare et al., 2024).

These 3D model representations for DTs can be categorized according to their target application, which determines requirements related to geometry, semantics, interoperability, and scalability.

(i) *Infrastructure and Facility Management:* models linked to asset inventories, maintenance records, and space utilization data, supporting inspection planning, asset lifecycle management, and integration into operational DT platforms.

(ii) *Monitoring and Structural Analysis:* models integrated with finite element models (FEM), structural health monitoring systems, and Internet of Things (IoT) sensors, enabling continuous condition assessment, damage detection, load and deformation analysis, and long-term safety or resilience evaluation.

(iii) *Urban Planning and GIS:* city-scale models using formats such as CityGML, CityJSON, or 3D Tiles, supporting zoning analysis, infrastructure planning, shadow studies, and web-based public visualization.

(iv) *Heritage Documentation:* high-fidelity geometric and textured reconstructions of cultural heritage assets, aimed at preservation, restoration, archiving and virtual museum or dissemination experiences.

(v) *BIM Integration:* import of reconstructed geometry into IFC-compliant Building Information Models (BIM), enriched with parametric and semantically structured elements to support design coordination, renovation processes, and facility management.

(vi) *AR/VR Visualization:* optimized and visually realistic models designed for real-time, immersive environments, used in presentations, stakeholder engagement, training, and safety simulations.

(vii) *Energy and Performance Analysis:* simplified yet physically consistent volumetric models incorporating materials and boundary conditions, enabling solar, daylighting, thermal comfort, and energy performance simulations.

2.2.1 Digital Twin domain support

DTs applications can also be described across different fields of application and generation, each adding increasing layers of information and functionality to the digital representation:

- *Geometric Digital Model:* A 3D representation of the building primarily used for visualization and basic spatial understanding, generally a mesh-based 3D model.
- *Semantic Digital Model (BIM-like DT):* A geometric model enriched with semantic information on spaces, building elements, materials and systems, supporting coordination, documentation and asset management.

- *Data-Driven Operational Digital Twin:* A semantic model connected to sensors and external databases, enabling real-time or near–real-time monitoring of energy consumption, indoor environmental conditions, and equipment performance.
- *Simulation and Predictive Digital Twin:* An operational DT coupled with analytical and control models, allowing advanced simulations, *"what-if"* scenario analysis, optimization processes, and predictive maintenance strategies.

2.2.2 Level of Development or maturity of DT

As shown in the previous sections, a fully resolved DT requires the integration of different 3D models. A DT should be purpose-defined and developed only to the level of fidelity requested for its intended use. To describe how structure and data–geometry integration evolves, different authors propose maturity scales, framed in various ways. In (Hu et al., 2023), maturity progresses through six levels that describe how advanced and connected a DT is:

- *Level 1 – Replicability*: a faithful digital copy of the physical asset.
- *Level 2 – Connection*: domain models are linked to the replica to simulate processes.
- *Level 3 – Synchronization*: real-time data streams keep the digital and physical assets aligned.
- *Level 4 – Interaction:* bidirectional control enables remote actuation and feedback.
- *Level 5 – Automation:* autonomous decision-making and learning optimize operations.
- *Level 6 – Interoperability:* multiple DTs collaborate across platforms for joint optimization.

In short, maturity increases from basic representation to intelligent, interconnected ecosystems.
On the other hand, (Mousavi et al., 2024) condenses maturity into three tiers: DTL1 (Monitoring), focused on sensing and visualization; DTL2 (Simulation), enabling what-if analysis and process modelling; and DTL3 (Predictive Maintenance), where analytics anticipate failures and support informed intervention. In particular:

- *DT Level 1:* real-time sensor integration for monitoring, visualization, and diagnostics.
- *DT Level 2:* dynamic simulation, forecasting, and optimization (also from historic data).
- *DT Level 3:* analytics for failure prediction and strategic building management, contingent on historical data availability.

Given this variety, it is essential, just as for any digital model, to define the purpose of the DT from the outset. The intended use should drive the required level of development, the amount of information to be included and the assignment of responsibilities, since each purpose implies a specific DT typology and maturity level.

2.3 Required steps for a full DT: an overview

As illustrated in the proposed framework, the generation of energy-oriented DTs relies on interdependent workflows addressing data acquisition, geometric reconstruction, model structuring, semantic enrichment, and simulation-driven analysis. Due to this intrinsic complexity, a clear distinction between process stages and model abstractions is required to ensure methodological coherence.
The Scan-to-EDT framework therefore adopts a multi-level modeling strategy, in which geometric, topological, and semantic representations are progressively constructed and refined. Early stages focus on data-driven geometry reconstruction from heterogeneous sensing sources, while later stages emphasize topological structuring, knowledge integration, and interoperability with energy simulation engines.

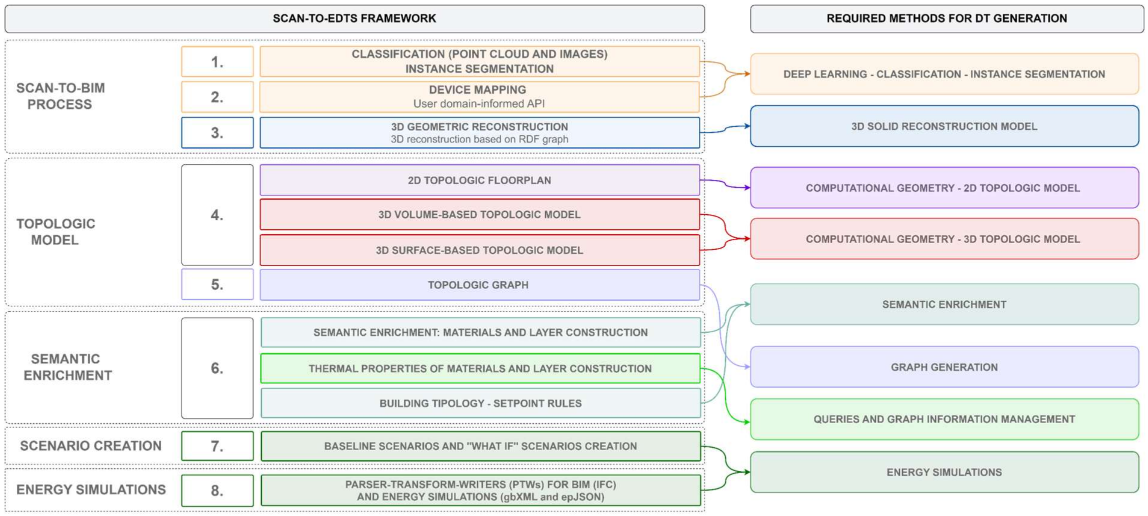


Fig. 11. Layered architecture of the Scan-to-EDTs and the involved modeling and analysis methods.

The following sections review the principal methods and technologies underpinning each workflow stage, with particular emphasis on DT generation, multi-source data integration, and energy performance analysis, as illustrated in Figure 11.

In this context, starting from Fig. 7, Fig. 12 provides an overview of the correspondence between the key topics addressed within the Scan-to-EDTs framework and the respective sections of this chapter, which collectively present the state of the art for each processing step.

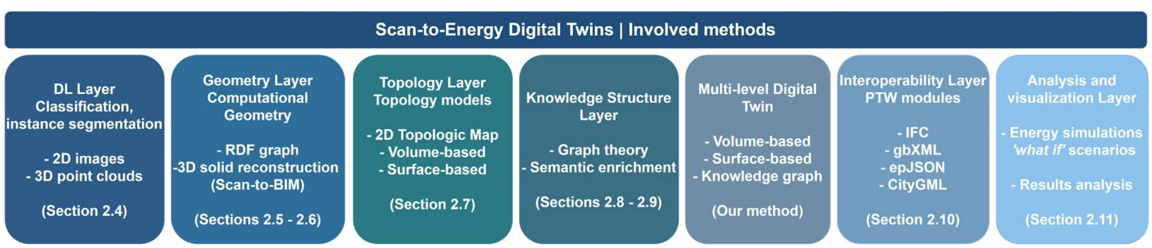


Fig. 12. Scan-to-EDTs framework, the related steps involved and corresponding chapter sections.

2.4 Semantic segmentation, classification, instance segmentation

Terminology for classification, semantic segmentation, and instance segmentation in ML and DL is often used inconsistently across application domains and purposes.

In this section, these tasks are defined consistently. Each applies to both images and point clouds, serving distinct functions and, as discussed later, combining outputs from both modalities can further improve accuracy.

To define them clearly the different definitions are reported below (Figure 13):

- *Classification:* the network predicts one label per input unit. For images, the unit can be the whole image or a cropped region (e.g., "grass," "cat", "tree"). For point clouds, it can refer to the entire scan, a zone, or a single component (e.g. in constructions, "bridge," "beam," "column").
- *Semantic segmentation:* For images, each pixel is labelled according to its category (e.g., wall, ceiling, window, furniture). For point clouds, each point receives a class label, distinguishing structural or functional elements such as slabs, pillars, pipes, or ducts.

- *Instance segmentation:* For images, the network identifies and separates individual objects of the same category (e.g., Column 1, Column 2, Beam 1, Beam 2). For point clouds, in the same way, it isolates each distinct 3D object instance (e.g., Column A, Column B, …).

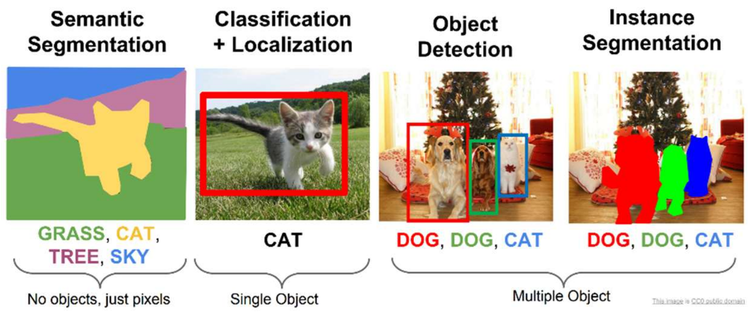


Fig. 13. Differences between semantic segmentation, classification and instance segmentation[1].

2.4.1 Point cloud-based classification and semantic segmentation

Semantic segmentation of point clouds can be performed through multiple approaches, primarily differing in their representation and processing of 3D data (Fig. 14). Point-based methods learn directly from raw point clouds, operating on unordered (x, y, z) coordinates to preserve fine geometric details. Pioneering models such as *PointNet* (Qi et al., 2017a) and *PointNet++* (Qi et al., 2017b) introduced direct learning on point sets and hierarchical feature extraction, while later works like *DGCNN* (Wang et al., 2019) used dynamic graph convolutions to capture local neighborhood relations.

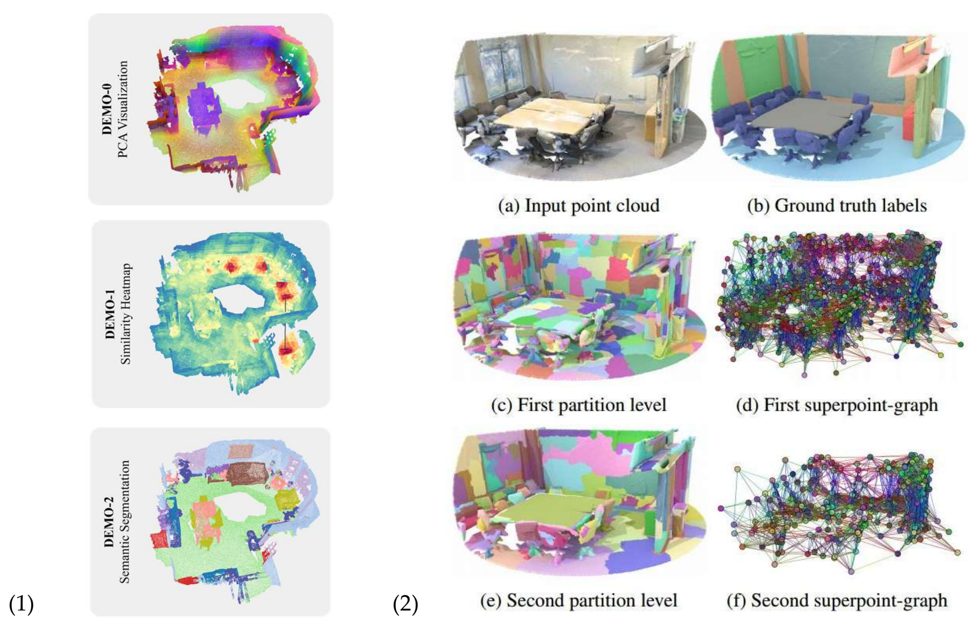


Fig. 14. Some examples of Transformers for 3D point cloud semantic segmentation and classification: (1) Point Transformer V3 (Wu et al., 2024) and (2) Superpoints Transformer (Robert et al., 2023).

[1] Karpathy, A., Johnson, J., 2024. Cs231n convolutional neural networks for visual recognition-transfer learning. https://cs231n.github.io/.

More recent architectures such as *KPConv* (Thomas et al., 2019), *PointCNN* (Li et al., 2018), and transformer-based networks (*Point Transformer* (Wu et al., 2024), *PointNeXt* (Qian et al., 2022), *Superpoint Transformer* (Robert et al., 2023) further improved local–global feature aggregation and scalability.

Voxel-based methods discretize the 3D space into voxels, enabling the use of 3D convolutions for structured learning. Early models, like *VoxelNet* (Zhou and Tuzel, 2017), laid the foundation for voxel-based processing, followed by *SparseConvNet* (Graham et al., 2017) and *MinkowskiNet* (Choy et al., 2019), which introduced sparse convolution operations for efficiency.

Extensions, such as *3D U-Net*, enhanced spatial resolution and large-scale segmentation performance. Projection-based methods convert point clouds into 2D representations (e.g., range images or multi-view projections), allowing the use of mature 2D CNN architectures. Finally, hybrid and graph-based methods integrate multiple representations or use graph neural networks (GNNs) to capture both geometric and topological relationships. Examples include *SPGraph* , *Point-GNN* (Shi et al., 2020) and hybrid fusion frameworks like *3D-SIS* (Hou et al., 2019), which combine voxel, point, and image features for richer contextual understanding.

Overall, research has progressed from rigid voxel or projection-based grids to flexible point and hybrid frameworks, emphasizing scalability, geometric precision, and applicability to complex real-world tasks such as Scan-to-BIM, autonomous driving, and DT creation.

2.4.2 Image-based classification and semantic segmentation

Semantic segmentation on images has evolved through several key methodological families that have advanced accuracy, scalability and contextual understanding. Fully Convolutional Networks (FCNs) introduced pixel-level prediction using convolutional layers, while encoder–decoder architectures such as SegNet (Badrinarayanan et al., 2016) and U-Net (Ronneberger et al., 2015) restored spatial resolution through symmetric upsampling, making them effective for structured scenes. More recently, transformer-based architectures, such as SegFormer (Xie et al., 2021), Mask2Former (Cheng et al., 2022), and Segmenter (Strudel et al., 2021), unified global context and local detail, achieving state-of-the-art results on COCO (Lin et al., 2014), Cityscapes (Cordts et al., 2016) and ADE20K (Zhou et al., 2019). Furthermore, multi-task DL enables 3D reconstruction and semantic annotation through image segmentation, transforming single RGB images into BIM-compatible 3D representations via domain adaptation (Erişen et al., 2025).

In parallel, real-time and lightweight models like Fast-SCNN (Poudel et al., 2019) and YOLO-seg (Khanam and Hussain, 2024) enable efficient deployment in navigation, robotics, and mobile applications.

The latest advances include foundation and vision–language models such as Segment Anything Model (SAM) (Kirillov et al., 2023) and GroundingDINO (Liu et al., 2023), which bring unprecedented generalization and multimodal capability. SAM, built on a Vision Transformer (ViT) backbone and trained on the SA-1B dataset (Kirilov et al., 2023), performs prompt-based, class-agnostic segmentation in real time using points, boxes, or text inputs. GroundingDINO extends this paradigm by combining visual detection with natural language grounding, enabling open-vocabulary object detection and segmentation. Together, these models are increasingly applied in remote sensing, built-environment analysis, and Scan-to-BIM workflows, bridging 2D and 3D domains for scalable, interactive and generalizable DTs generation.

2.4.3 Object detection in challenging environments

Detecting small or occluded indoor objects is particularly difficult when they blend into their surroundings, such as wall-integrated radiators, recessed luminaires, or embedded sensors (Dai et al., 2025). In these scenarios, the objective often shifts from precise localization to determining whether an object is present within

a given Region of Interest (ROI). Region-based Convolutional Neural Networks (R-CNNs) support this paradigm by proposing candidate regions that are then classified (Lin et al., 2024).
Multi-scale feature extraction mitigates scale variation across objects of different sizes, while context-aware recognition leverages spatial and semantic cues to infer likely object presence. Region Proposal Networks (RPNs) generate candidate object locations and bounding-box–based loss functions, such as Intersection over Union (IoU), are used to assess detection quality. Anchor boxes encode prior assumptions on object size and aspect ratio, and probabilistic scoring functions provide confidence estimates that help handle occlusions and partial visibility. Comparable strategies have been successfully applied in other complex environments, including forestry (Fol et al., 2024) and agricultural harvesting (Shi et al., 2025).
More recently, a promising 3D Large Language Model (LLM) has been proposed in (https://github.com/manycore-research/SpatialLM) capable of processing point cloud data to produce structured 3D scene interpretations, enabling richer, higher-level analysis of complex environments.

2.4.4 Late fusion approaches

Late fusion processes each modality with its own backbone and combines only high-level features or decisions at the final stage (Yang et al., 2024). This method preserves modality-specific inductive biases and reduces cross-modal interference. It is widely used for heterogeneous sensors, e.g., RGB, LiDAR, GNSS/IMU (Ren et al., 2024), to improve robustness in challenging conditions such as tunnels (Ji et al., 2022), cluttered indoor environments, and dense vegetation (Samadzadegan et al., 2024).
In urban scale building segmentation, late fusion often aggregates predictions from high-resolution satellite imagery, DSMs, and ancillary layers such as OpenStreetMap, enhancing instance delineation and boundary sharpness. In 3D reality-based surveying, parallel 2D-image and 3D-point-cloud branches are common: per-view 2D semantic segmentations are projected onto the point cloud and fused with point-cloud network outputs just before the final decision.
From a fusion standpoint, two variants are typically applied in weighting multiple point cloud fusion:

- *Non-weighted fusion:* uniform averaging or majority voting over modality scores, assuming equal reliability.
- *Weighted fusion:* confidence-aware or learned weights that emphasize texture-dominant regions (image cues) or geometry-dominant regions (point-cloud cues). Weighted late fusion generally yields more stable, consistent predictions across scenes than uniform schemes.

2.5 Computational geometry strategies for 3D reconstruction

Computational geometry approaches (de Berg et al., 2008) for 3D building reconstruction can be characterized along multiple complementary dimensions, including semantic granularity, representation models, and computational strategies. These dimensions are not mutually exclusive; rather, they jointly influence reconstruction accuracy, scalability, and suitability for DT–oriented workflows.
From the perspective of semantic granularity, reconstruction methods differ in how explicitly individual building elements are distinguished:

- *Per-class reconstruction* operates on automatically labeled semantic classes (e.g., walls, floors, roofs), producing dense class maps without explicit instance separation. This approach is well suited for extracting building footprints, generating LOD1–LOD2 city models, and supporting coarse semantic understanding.
- *Per-object (instance-based) reconstruction* introduces instance differentiation, typically through instance segmentation techniques. By assigning unique identifiers to individual components, these approaches enable object-level geometric fitting (e.g., oriented bounding boxes and parametric

primitives) and support BIM element generation, construction-phase comparison, and change detection, albeit at the cost of higher computational demands and more complex processing pipelines.

When considering the underlying representation model, reconstruction approaches differ in how geometry and relationships are encoded:

- *Graph-based reconstruction* (Sun et al., 2025b) represents buildings as networks of elements and relationships, using graph neural networks or combinatorial optimization to jointly infer geometry, topology, and semantics. This enables applications such as indoor scene reasoning, multi-story reconstruction, and BIM graph generation aligned with ontologies such as IFC, BOT, and Brick.
- *Procedural and rule-based reconstruction* (Sun et al., 2025a) relies on grammars, parametric rules, or inverse procedural modeling to derive structure from repetitive or regular patterns, making these methods particularly effective for façade reconstruction and generative urban modeling.

From the standpoint of computational strategy, reconstruction methods vary in their level of automation and adaptability:

- *Data-driven and learning-based reconstruction* employs DL models, including convolutional networks (CNN), transformers, implicit representations, and neural radiance fields (NeRFs) (Mildenhall et al., 2020), to infer geometry or occupancy directly from sensor data. While offering high automation and fine detail, these methods often produce non-editable and semantically opaque models that require additional processing for BIM or DT integration.
- *Hybrid and multi-modal reconstruction* combines complementary sensors and methods (e.g., LiDAR and photogrammetry, deterministic fitting, and learned components) to improve robustness, completeness, and semantic richness, particularly in cluttered or occluded environments.
- *Incremental and adaptive reconstruction* uses techniques such as SLAM (Szrek et al., 2024), incremental mesh fusion, and streaming point cloud processing to update models over time, enabling real-time construction monitoring, as-built verification, and reconstruction in dynamic environments.

Together, these complementary dimensions provide a structured lens for analyzing existing 3D reconstruction approaches and for assessing their suitability for DT–oriented building workflows.

2.5.1 Geometric Representation

Geometric representations define how 3D models are computed, visualized, and stored digitally. The choice of representation fundamentally affects the precision, semantic richness, and computational efficiency of reconstruction workflows. Common representations include:

- *Point clouds:* raw 3D data captured from LiDAR or photogrammetry, consisting of unstructured collections of coordinates with optional attributes (color, intensity, normals).

Surface-Based Representations:

- *Mesh models:* triangulated or polygonal surfaces representing building geometry, typically encoded as vertices, edges and faces (Maalek et al., 2019).
- *Boundary representation (B-Rep):* exact solid models defined by topologically connected faces, edges, and vertices, commonly used in CAD and BIM systems for precise geometric modeling (Kada and McKinley, 2009).

Parametric and Semantic Representations:

- *Parametric (BIM) models:* semantic, object-based representations with rich attributes such as thickness, material properties, thermal characteristics, and construction specifications. These often employ solid modeling kernels or B-Rep representations internally, with elements organized according to standards like IFC (Industry Foundation Classes).

Volume-Based Representations:

- *Volumetric or voxel models:* volumetric or voxel-based models require substantial prior knowledge and are therefore rarely used for direct 3D reconstruction (Tang et al., 2010). They are instead mainly adopted for numerical analysis and simulation, such as in Computational Fluid Dynamics (CFD) tools like OpenFOAM (OpenCFD Ltd) and finite element frameworks such as FEniCS (Logg et al., 2011), where volumetric discretization is essential to solve governing physical equations.
- *Constructive Solid Geometry (CSG):* solid models built through Boolean operations (union, subtraction, intersection) applied to primitive shapes (spheres, cylinders, boxes). Widely used in CAD systems and procedural modeling workflows for representing complex geometry through hierarchical combinations of simple forms.

Each representation offers distinct tradeoffs: point clouds preserve raw measurement fidelity but lack structure; meshes balance detail and efficiency for visualization; B-Rep and CSG provide exact geometry suitable for manufacturing and engineering analysis; BIM models add semantic layers essential for facility management; and voxel grids enable physics-based simulation at the cost of memory and resolution constraints.

2.5.2 Explicit and Implicit Modelling

From a computational geometry perspective, reconstruction approaches can be broadly classified as (i) explicit, (ii) implicit, or (iii) hybrid, each offering different trade-offs in terms of geometric precision, automation, and topological flexibility. Figure 15 illustrates how these strategies progressively bridge reconstructed models and downstream engineering applications.

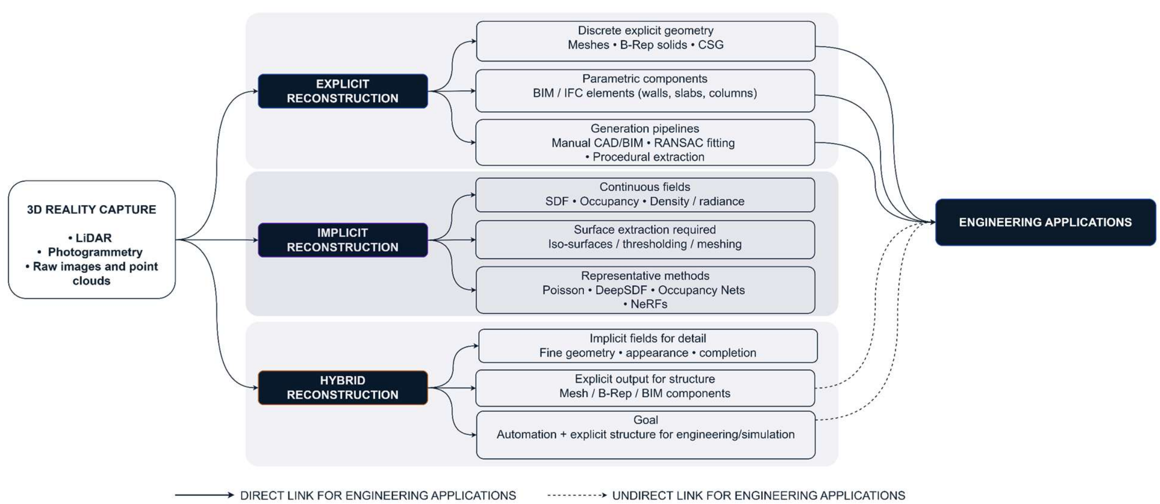


Fig. 15. Types of geometric modeling representations derived from 3D (and 2D-derived) reality capture data; only explicit and selected hybrid reconstructions are suitable for engineering applications.

Explicit modeling techniques encode geometry in structured and interpretable forms, enabling the generation of models that are directly compliant with engineering requirements. In contrast, implicit representations are well suited for visualization and continuous surface modeling, but generally lack the explicit structure required for engineering-oriented applications. Hybrid approaches combine both paradigms; however, only those grounded in B-REP, CSG, or BIM representations provide effective compatibility with engineering workflows.

- *Explicit Methods* describe geometry via discrete surfaces or volumes (meshes, B-REP solids, CSG trees, parametric BIM elements such as IFC walls, slabs, and columns). Models may be created manually in CAD/BIM, semi-automatically via surface fitting and segmentation (e.g., RANSAC (Hemmer et al., 2024) plane extraction), or fully automatically through component extraction and procedural pipelines. Recent workflows derive structural elements directly from LiDAR or photogrammetric point clouds at urban scale (Nan and Wonka, 2017; Chen et al., 2024). Explicit models are interpretable, precise, and well suited to engineering analysis, fabrication, and regulatory workflows.
- *Implicit Methods* represent geometry as continuous fields (occupancy grids, signed distance functions, density fields), with surfaces defined as level sets or thresholds. Classical methods, such as Poisson surface reconstruction (Kazhdan et al., 2006) and neural implicit models, such as DeepSDF (Park et al., 2019), Occupancy Networks (Mescheder et al., 2019), or NeRFs (Mildenhall et al., 2020), reconstruct buildings from images or point clouds, yielding smooth, topologically flexible surfaces that handle complex or incomplete data, but typically require post-processing to obtain explicit geometry.
- *Hybrid Approaches* integrate implicit representations for robust shape inference and fine geometric detail with explicit outputs, such as meshes or parametric BIM components, to support structural modeling and simulation. By combining the automation and data tolerance of implicit methods with the semantic clarity, topological structure and interoperability of explicit models, hybrid workflows aim to bridge the gap between raw sensing data and simulation-ready DTs.

Model Type	**Geometry type**	**Representation**	**Primary Use Cases**	**Tools & Frameworks**	**Semantic Level**
BIM	Explicit	Parametric objects (walls, slabs, etc.)	Design, facility management, compliance	Revit, ArchiCAD, IFC	Very high: geometry + semantic attributes
Mesh	Explicit	Triangulated surfaces	Visualization, AR/VR, gaming	Blender, MeshLab, Unity	Low: geometry only
B-Rep/CSG	Explicit	Solid models via faces/edges or Boolean operations	CAD, manufacturing, precise engineering	Rhino, SolidWorks, FreeCAD	Medium: geometry + construction logic
Neural reconstruction	Implicit	Continuous fields (SDF, NeRF, Occupancy)	Photorealistic rendering, scene reconstruction	NeRFStudio, Instant-NGP, DeepSDF	Low: geometry & appearance only

Table 3. Comparison of geometric representation types for DT models, including geometry, semantics, use cases, and supporting tools.

Data storage and interoperability depend instead on the target application. BIM workflows typically rely on IFC or proprietary formats (e.g., Revit families), while urban-scale semantic modeling adopts standards such as CityGML or CityJSON.

For lightweight visualization and real-time rendering, formats such as OBJ, PLY, glTF or USD are commonly used. For analytical purposes, explicit geometric models are further transformed into domain-specific representations, including Building Energy Models (BEM) for thermal simulation, Finite Element

Models (FEM) for structural analysis, and Digital Surface or Elevation Models (DSM/DEM) for terrain and granular-flow analyses.

2.5.3 Level of Automation

Modeling strategies span a continuum of automation that affects productivity, scalability, and required expertise, ranging from fully manual to fully automatic workflows.

- *Manual approaches* rely on expert-driven extraction of sections and profiles from point clouds or meshes, followed by NURBS fitting and BIM element creation in tools such as Revit or Archicad. They provide high geometric fidelity and control but are time-consuming, labor-intensive, and operator-dependent, making them suitable for complex or heritage projects.
- *Semi-automatic approaches* automate selected steps using scripting or parametric tools (e.g., Grasshopper, Dynamo, Python), including denoising, meshing, primitive extraction, and rule-based element generation (Andriasyan et al., 2020). These workflows improve efficiency while retaining user oversight but remain constrained by predefined assumptions.
- *Automatic approaches* employ ML/DL models to process raw point clouds, images, or multimodal data for scene segmentation and geometry reconstruction, enabling scalable BIM generation. While efficient, they are sensitive to data quality and domain shift and typically require downstream validation and semantic refinement.

In practice, manual methods suit high-value or heritage projects, semi-automatic workflows balance control and throughput for standard developments, and fully automatic pipelines support large-scale digital twins where approximate, yet scalable models are acceptable.

2.5.4 From 3D model meshes to specific-purpose DT models

The most common geometric representations, including B-Rep for precise CAD and BIM modeling, Constructive Solid Geometry (CSG) for procedural or parametric design and mesh models for large scale 3D reconstruction and visualization, generally retain a low level of semantics after semi-automatic or automatic processing (Table 4).

Consequently, they must be further structured to accommodate rich and well-defined information such as object classes, relationships, and material properties in a consistent and machine-readable form.

To effectively structure 3D models for DT creation, it is essential to clearly define the primary purpose of the model. Table 4 defines the type of model required for each application.

Application / Analysis Type	Purpose	Typical Model Type Required	Examples / Tools	Level of Semantics
Indoor energy monitoring	Track real-time energy use, temperature, humidity, and comfort conditions	Parametric BIM models linked to sensor data (if possible), volumetric or Digital Twin environments	*Revit*, *IFC*, *Microsoft Azure DT*, *TwinCAT*, *Node-RED*	**High** — combines geometry, materials, and sensor semantics
Energy analysis	Evaluate energy performance, HVAC efficiency, and thermal comfort	Building Energy Models (BEM), volumetric, zone-based model	*EnergyPlus*, *IES VE*, *DesignBuilder*	**High** — includes materials, thermal zones, and usage patterns
Energy simulations	Perform dynamic simulations of heating, cooling, lighting, and ventilation	BIM-based BEM; volumetric models with physical parameters; B-Rep surface models prepared for simulation	*EnergyPlus*, *TRNSYS*, *OpenStudio*	**High** — rich semantic and physical attributes
CFD simulation	Simulate indoor airflow, temperature distribution, and pollutant dispersion	Volumetric or voxel models; surface meshes derived from CAD/BIM geometry	*OpenFOAM*, *SimScale*, *Ansys Fluent*, *FEniCS*	**Low to medium** — primarily geometric data and boundary conditions

Table 4. Model requirements and semantic levels for energy-related applications and analyses in DT environments.

Table 4 summarizes, in the energy domain, the model types required for condition monitoring (DTL1), energy or CFD-based scenario simulations (DTL2), and ML or DL towards a predictive maintenance (DTL3) replica. These models differ in purpose, data requirements, geometric encodings and the geometric types they represent.

To support all these operations within a single DT, a multi-level DT is then crucially required:

(i) *Geometric Diversity:* both volumetric models (for CFD, FEM, energy analysis) and surface-based B-Rep representations (for BIM and geometric reasoning) must coexist and be interoperable.

(ii) *Data Heterogeneity:* integration of heterogeneous data types and systems, including static geometric and semantic data (IFC, CityGML), dynamic sensor streams, simulation results, and predictive analytics outputs, within a unified data architecture.

(iii) *Workflow typology:* automated pipelines for data transformation, model synchronization, analysis execution, visualization, and decision support, often implemented using workflow engines, microservices, or knowledge graphs.

This multi-level approach enables DTs to serve simultaneously as real-time monitoring dashboards, simulation testbeds, and predictive maintenance platforms, each operating on geometrically and semantically appropriate model representations.

2.6 Scan-to-BIM approaches

Scan-to-BIM processes convert dense point clouds into semantically rich and interoperable BIM models, and automated BIM generation remains an emerging domain with few truly end-to-end solutions (Skrzypczak et al., 2022). Beyond geometric fitting, it must ensure semantic consistency, spatial reasoning and topological connectivity, bridging BIM's object centered solids (for example, walls and slabs) with space-centered workflows such as rooms, zones, and thermal boundaries. Traditional manual or semi-automatic pipelines based on segmentation, primitive fitting, and BIM modeling are accurate but slow and difficult to scale.

A typical Scan-to-BIM workflow can be broadly divided into four main stages (Bassier et al., 2019).

1. *Data acquisition.* A point cloud is generated using either image or range-based techniques.
2. *Segmentation.* The raw point cloud is partitioned into regions or clusters that exhibit similar geometric or radiometric characteristics, providing an intermediate structural organization of the data.
3. *Classification.* Each segmented region is assigned a semantic label (e.g., wall, floor, column), linking geometric information to domain-specific semantics.
4. *Reconstruction and modelling.* The labelled regions are transformed into a structured 3D model that integrates both geometry and semantic information, typically in a BIM-compliant representation.

In practice, point cloud data are commonly preprocessed through filtering, downsampling, or noise removal to reduce data size and improve computational efficiency (Shi et al., 2018). However, these operations may compromise data fidelity, and the quality of the resulting BIM models remains highly dependent on software tools and subjective operator decisions (Zhang et al., 2022).

For 3D reconstruction and modeling, many workflows (Pantoja-Rosero et al., 2024) first apply *scan-to-mesh* to derive objects from the point cloud or images and then rewrite geometries into BIM standards such as IFC (Perez-Perez et al., 2021). Scan-to-mesh converts point clouds directly into polygonal meshes, which can be semantically enriched using outputs from the classified point cloud.

This approach is useful when:

- geometry is complex or free-form, and parametric BIM recovery is brittle;

• rapid visualization, simulation, or collision checks are required; or
• BIM is needed later; mesh-to-BIM pipelines can, in fact, extract surfaces, convert them to B-Reps, map geometries into BIM and add semantic metadata.

In practice, these types of pipelines are common: *point cloud* → *scan-to-mesh* for fast, geometry-complete context, followed by targeted *scan-to-BIM* for elements that benefit from parametric control, such as walls, openings and structural components.

Recent DL-based methods have significantly improved the automation of semantic and instance segmentation, yet notable challenges remain, particularly in handling occlusions, complex geometries, and enforcing BIM-compliant topology and parametric constraints. End-to-end approaches such as Scan2BIM NET (Pérez Pérez et al., 2021) improve segmentation accuracy, NeRF to BIM (Hachisuka et al., 2023) addresses occlusions through neural radiance fields, and BIM Net++ (Campagnolo et al., 2023) enhances object detection. More comprehensive pipelines are proposed by (Bassier et al., 2020a), including dedicated wall reconstruction (Bassier et al., 2020b) and graph-based extensions (Bassier et al., 2024).

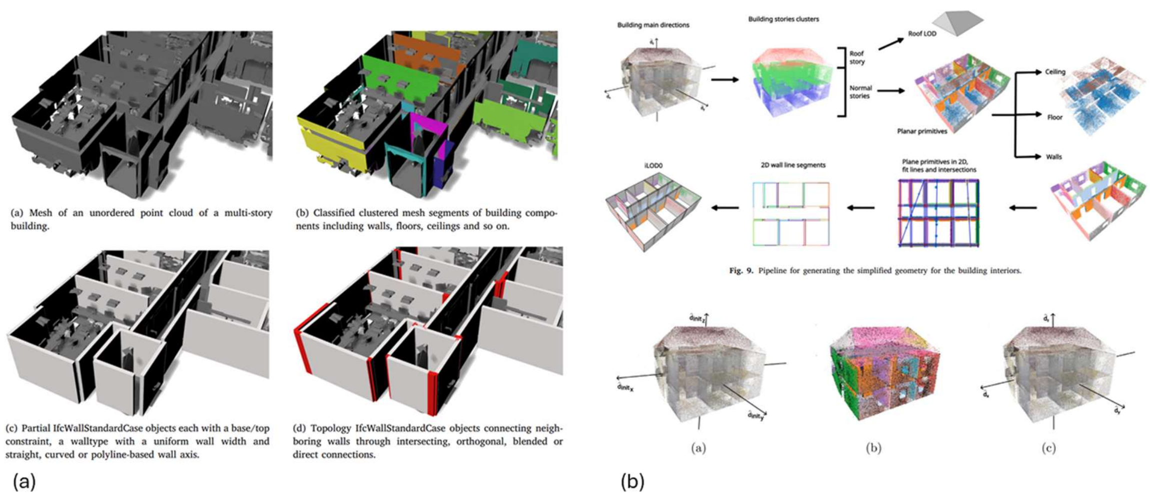


Fig. 16. Scan-to-BIM examples: (a) IFC wall reconstruction (Bassier & Vergauwen, 2019); (b) LOD4 building reconstruction with an image-based methodology (Pantoja-Rosero et al., 2025).

Alternative strategies include statistical models using reversible jump MCMC (Tran and Khoshelham, 2020) and computational geometry and edge extraction techniques (Li et al., 2024), which support structural monitoring (Jiang et al., 2022), feature evaluation (Özkan et al., 2024), and construction progress tracking (Kim, 2020).

Topologic BIM approaches (Daum et al., 2014) represent buildings using topological abstractions rather than purely geometric elements. Rooms are modeled as volumetric cells, walls and floors as boundary faces, and functional zones such as thermal or fire compartments as higher-level spatial entities. Rule-based modeling and topological queries, including adjacency, containment and connectivity, enrich these entities with domain-specific attributes while reducing geometric complexity.

At the same time, Topologic BIM maintains semantic consistency and interoperability with building performance simulation tools, IoT data streams and DT platforms, and remains compatible with established ontologies such as IFC, BOT, and Brick.

In contrast to early Scan-to-BIM outputs that focus on isolated geometry, Topologic BIM explicitly encodes spatial relationships as graphs, enabling the transition toward DTs that integrate geometry, topology, simulations and sensor data within a unified representation.
Building on this foundation, graph-based DTs extend static BIM environments by incorporating real-time sensor data, simulation results, and ontological reasoning within comprehensive knowledge graphs, enabling intelligent querying, predictive analytics, and adaptive control.

2.7 Topologic models

A topological model in 2D or 3D is a formal representation that encodes spatial entities as nodes, edges, faces, and volumetric cells, together with their incidence, adjacency, containment, and connectivity relations, independently of metric geometry (Fig. 17).
Such a model provides a dimension-independent spatial abstraction, formalizing objects as a graph of primitives and their relations and thereby enabling robust spatial reasoning and analysis beyond purely geometric descriptions.
Within this paradigm, topological building models explicitly encode spatial relationships, such as adjacency, containment, and connectivity, rather than relying solely on geometric form (Arroyo Ohori et al., 2015). This explicit relational structure supports advanced spatial reasoning, navigation, and functional analysis that purely geometric models cannot achieve, making topology a foundational component for advanced Digital Twins, performance-driven simulations, and urban-scale planning.

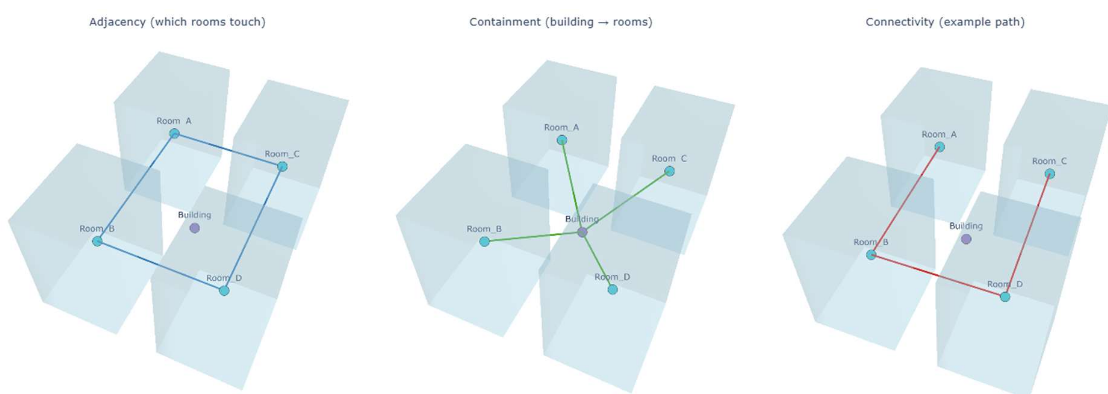


Fig. 17. Adjacency, containment, and connectivity relations in a topological building representation.

2.7.1 2D Topologic models

Recently, a growing number of DL–based methods for 2D floorplan reconstruction have been proposed. Among them, RoomFormer (Yue et al., 2023) represents one of the most structured and end-to-end approaches. RoomFormer introduces a single-stage transformer-based architecture that reconstructs 2D floorplans directly from 3D point clouds by operating on a density map and predicting a variable-length set of room polygons.
Unlike earlier pipelines that rely on heuristic intermediate representations, such as corner detection, wall extraction, or semantic segmentation, RoomFormer jointly infers room geometry and topology.
The model employs a hierarchical query mechanism, consisting of room-level queries that represent individual polygons and corner-level queries that encode ordered vertex sequences.
This design enables the direct prediction of geometrically consistent polygonal rooms, ensuring correct corner ordering and closed shapes within a unified end-to-end framework. Other learning-based

reconstruction approaches include HEAT (Chen et al., 2022), which employs an attention-based neural architecture to capture long-range structural dependencies for planar layout reconstruction, and the work of (Van Engelenburg et al., 2024), which introduces a large-scale benchmark for topologically structured floor plan generation of multi-apartment residential buildings.

More recently, FRI-Net (Xu et al., 2024) proposes an alternative paradigm for 2D floorplan reconstruction from 3D point clouds by adopting a room-wise implicit representation rather than explicit corner-, wall-, or box-based regression. The network encodes each room into a latent embedding using a transformer-based encoder and decodes the embedding into a polygonal layout by predicting boundary lines, which are subsequently assembled into coherent room geometries. This implicit formulation yields more regular and topologically consistent polygons, resulting in improved reconstruction accuracy across multiple benchmarks.

2.7.2 3D Topologic models

Topological modeling plays a central role in energy analysis and digital twin development by enabling watertight, analysis-ready representations that are independent of purely geometric detail. By explicitly encoding incidence, adjacency, containment and connectivity relationships, topological models support robust volumetric computations, spatial reasoning, and the reliable extraction of energy-relevant parameters such as enclosed volumes, envelope surfaces and openings, which directly influence energy simulations and performance assessments.

At the building scale, topology is commonly embedded in boundary representations of solids and surfaces, where buildings are modeled as closed volumes composed of consistently oriented faces (Boguslawski et al., 2011; Ledoux and Meijers, 2011). This topological consistency enables the derivation of thermal zones, envelope components and adjacency relations required for Building Energy Models. Within BIM environments, these principles were formalized by (Borrmann and Rank, 2009) through a three-dimensional spatial query framework based on the nine-intersection model, supporting predicates such as within, touch, and overlap.

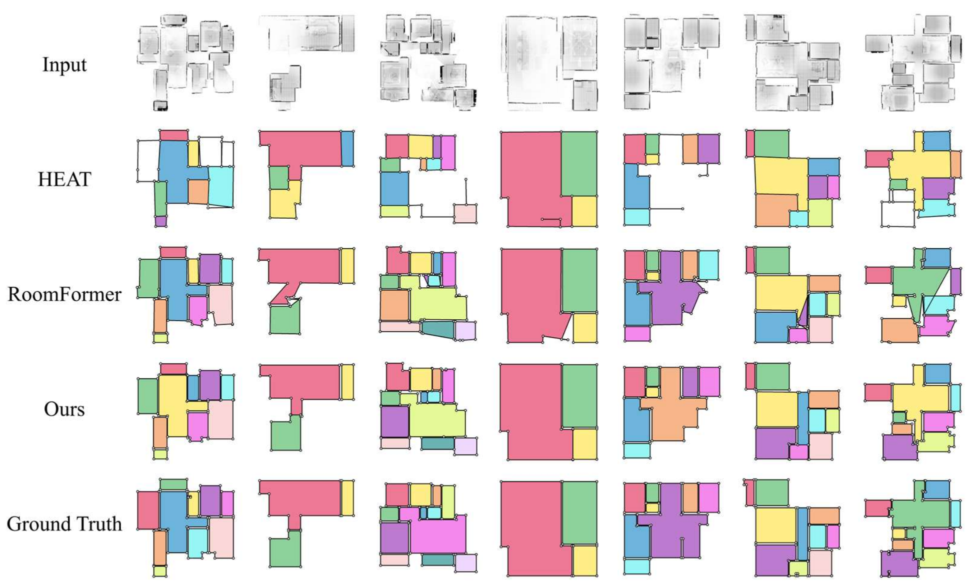


Fig. 18. Results of different networks for floorplan reconstruction on Strusture3D dataset (Xu et al., 2024). The 'ours' in the image is referred to (Xu et al., 2024) method.

By organizing vertices, edges, faces, and volumetric cells into graph or cell complex structures, topological BIM approaches enable rule-based reasoning, partial model extraction, and direct coupling with simulation workflows.

At the urban scale, similar principles underpin City Information Modeling (CIM), where topology supports accessibility, connectivity, and multi-building system representations. However, urban energy simulation requires additional topological constraints, including explicit volumetric definitions, surface semantics, and energy-related attributes. To address these requirements, advanced data structures such as the Dual Half Edge model (Boguslawski et al., 2011; Boguslawski and Gold, 2016) introduce a combined primal dual representation of non-manifold three-dimensional cell complexes, enabling explicit topology to be integrated with CityGML-like geometries. Across both scales, topological models bridge geometry and semantics by enabling queries to operate directly on adjacency structures rather than costly geometric intersection tests. This naturally leads to graph-based DTs, in which buildings and urban elements are represented as nodes, spatial and functional relationships as edges, and semantic, simulation and sensor data as attributes. Such representations provide a robust foundation for interoperable, ready analysis and adaptive DTs that extend beyond static three-dimensional models.

Finally, a key frontier lies in the automatic generation of topologically compliant three-dimensional models for energy and structural simulation using informed neural networks. SYNBUILD 3D (Mayer et al., 2025) addresses this challenge by introducing the first large scale synthetic dataset of residential buildings at LoD4, representing a major step toward deep learning-based generation of topological building models. The dataset comprises over 6.2 million multimodal instances, including fully integrated interior and exterior wireframes, semantic labels for rooms, doors, and windows, floor plans, and LiDAR-like roof point clouds (Fig. 19).

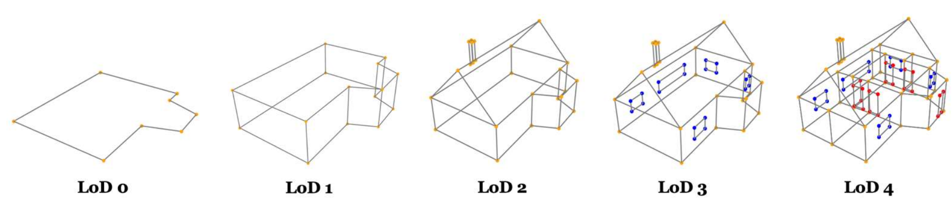


Fig. 19. LOD generation from (Mayers et al., 2025) in the DL-based SYNBUILD-3D approach.

By combining procedurally generated exteriors, AI-generated floor plans, and detailed semantic annotations into unified, topologically consistent three-dimensional models, SYNBUILD 3D fills a critical gap in the building modeling community and enables research on multimodal generative models, realistic interior and exterior reconstruction, and downstream architectural and simulation tasks at unprecedented scale.

2.8 Model enrichment: Information Loading Dictionaries and external libraries

Model enrichment augments reconstructed or topologically structured models with semantic, physical and operational information using Information Loading Dictionaries (ILD) and external domain libraries.

These mechanisms map geometric or topologic entities such as cells, faces, and edges to attributes from IFC property sets, BOT and Brick ontologies, material libraries, and performance datasets, transforming raw geometry into machine-interpretable representations.

Reference	Type	Main Enrichment Contribution
Belsky et al., 2016	BIM	Rule-based semantic enrichment of BIM elements, adding classifications, system grouping, and metadata to improve interpretability.
Bienvenido-Huertas et al., 2019	BIM (H-BIM)	Semantic labeling and classification of heritage elements using AI (J48), linking geometry to heritage-specific attributes.
El-Mekawy et al., 2011	CIM	Integration of BIM and GIS semantics to enrich city models with attributes supporting urban planning and 3D GIS analysis.
Isikdag & Zlatanova, 2009	BIM–GIS	Enrichment through BIM–GIS conversion schemas, enabling exchange of semantic and spatial information between building and city models.
Malhotra, 2023	GIS-CityGML	Semantic augmentation of city models for energy and environmental analysis, adding material, usage, and boundary-related attributes.

Table 5. Overview of representative studies on semantic enrichment for BIM, GIS, and city-scale models.

By attaching metadata including thermal properties on partition faces, equipment classes in mechanical spaces, and sensor types in monitored rooms, enrichment produces interoperable models suitable for simulation, compliance checking, facility management, and DT workflows. Across BIM and CIM contexts, enrichment methods apply rule-based inference, ontology-driven annotation, machine-learning classification, and BIM–GIS integration to introduce missing semantics, classifications, and spatial relationships, making models analysis-ready and operationally meaningful.

Table 5 reports the important works and their features.

These works illustrate key enrichment strategies: (Belsky et al., 2016) use rule-based inference to add semantics to BIM elements; (Bienvenido-Huertas et al., 2019) apply ML for automatic H-BIM labeling; (El-Mekawy et al., 2011) unify BIM and GIS semantics at city-scale; (Isikdag and Zlatanova, 2009) propose a conversion framework preserving BIM–GIS information; while (Malhotra, 2017) enriches city models with energy-related attributes for environmental analysis.

2.9 Graph Theory

Graph theory offers a rigorous and unifying framework for representing, analyzing, and synchronizing the complex relationships that underpin digital representations of the built environment. In graph theory, spatial systems are commonly represented using:

(i) *undirected graphs* (for mutual adjacency),
(ii) *directed graphs* (for hierarchical or flow relations), and
(iii) *labeled* or *attributed graphs* (where nodes and edges carry semantic properties), depending on the type of relationships being modeled.

In this perspective, a building is modeled as a graph, where nodes represent entities such as spaces, elements, and systems, and edges encode spatial, functional and dependency relationships, including adjacency, connectivity and containment (van Treeck and Rank, 2004; van Treeck and Rank, 2007).

This abstraction provides a bridge between geometry, semantics, and topology, enabling consistent interoperability across heterogeneous data sources, formats, and Information Loading Dictionaries (ILDs). Most AEC data structures, such as IFC, gbXML and CityGML, are inherently graph-based and often organized as hierarchical or tree-like structures (e.g., *building* → *storey* → *space* → *element*). These hierarchies can be extended into heterogeneous multilayer graphs that connect geometric, semantic, and physical domains, including MEP networks, sensor data, and structural systems. Recent work, such as IFC-graph, further demonstrates how graph-based representations can facilitate efficient building information access and querying (Zhu et al., 2023).

Latest DT workflows increasingly adopt knowledge graph architectures to manage these interconnections (Fig. 20). Labeled Property Graphs (LPGs), as implemented in graph databases like Neo4j (Neo4j, Inc., 2024), support fast, entity-centric traversal and relationship analytics, while RDF/OWL (El Idrissi et al., 2022) models enable rich semantic representation and automated reasoning based on established ontologies.

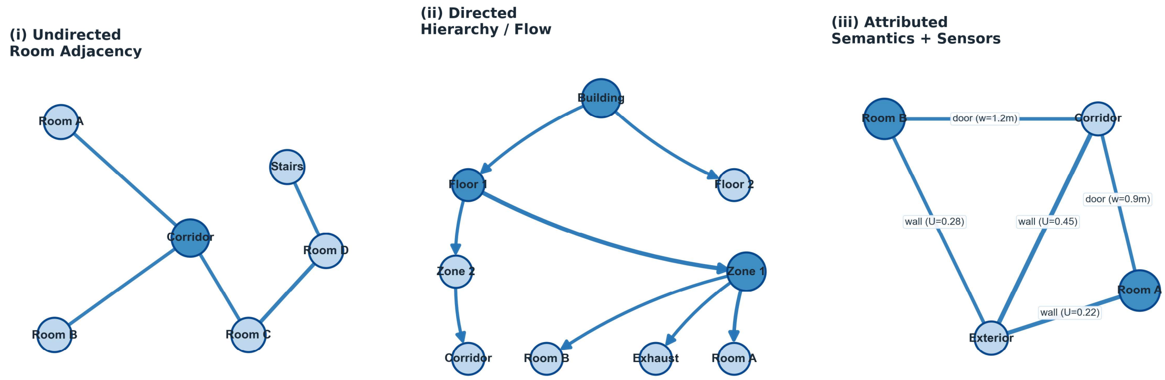


Fig. 20. Undirected, directed, and attributed graph representations of building topology.

Domain ontologies, such as the Building Topology Ontology (BOT), explicitly capture building relationships, and alignment or linear-referencing frameworks extend this approach to infrastructure networks and large-scale urban systems. Advanced techniques, including subgraph isomorphism and pattern matching, support the identification of recurring design motifs and dependency structures, while federated and fragmented graph storage enable distributed management of multi-domain datasets.

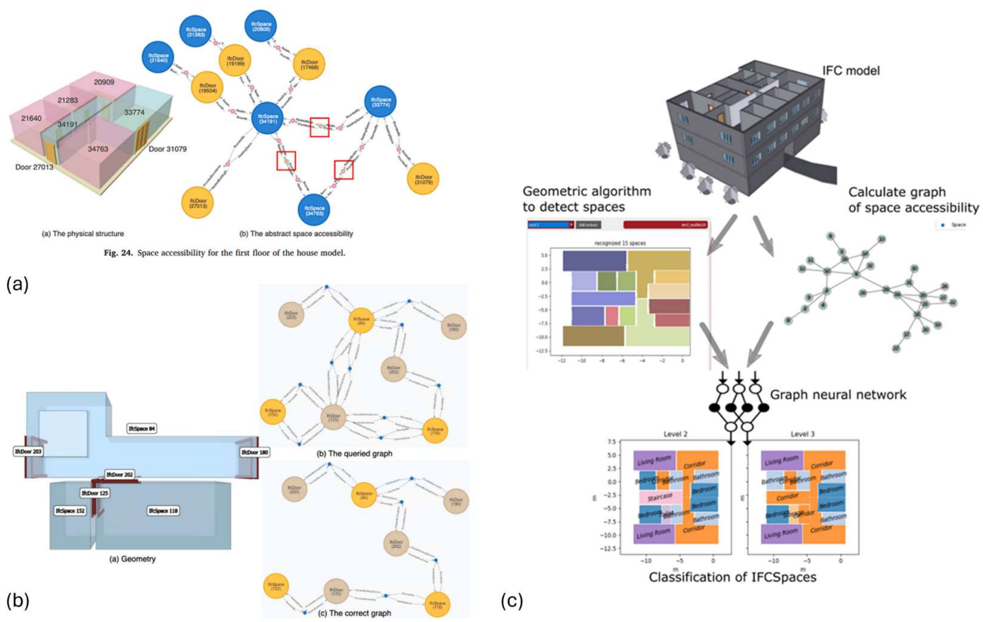


Fig. 20. Graph-based methods for indoor space modeling and analysis. (a) Zhu et al. (2023). (b) Zhu et al. (2025). (c) Burusz et al. (2025).

High-performance graph platforms, such as Neo4j (Neo4j, Inc., 2024) with composite databases, now allow cross-domain querying and synchronization at scale, bridging BIM, GIS, and IoT datasets within unified DT environments.

Graph-based approaches also integrate smoothly with relational-spatial systems like PostgreSQL/PostGIS (https://postgis.net/), where geometric operations (e.g., spatial adjacency or intersection) can be used to generate or validate graph edges, thereby linking geometric and semantic dimensions within a single analytical workflow. This convergence supports rich querying, reasoning, and change propagation, ensuring that modifications in one domain (geometric, semantic, or operational) are consistently reflected across others. Overall, graph theory provides a mathematical backbone for modern building data management, enabling interoperability and dynamic synchronization across BIM, GIS, IoT and DT frameworks.

2.10 Interoperability: models for simulations

Models for simulation differ in purpose, derivation, and data structure (Table 6) and may be geometric, semantic, topologic or physical.

Geometric models capture spatial form using B-Reps, meshes, or parametric solids; semantic models add material, functional and system attributes required for energy, structural or occupancy analyses; topologic models encode adjacency and connectivity for studies of circulation, accessibility, or energy flow; and physical or analytical models translate this information into computational structures such as BEM, FEM, or CFD grids.

A mature DT typically integrates several model types within a unified framework, selecting the appropriate representation for each analytical task. These models may be derived manually in CAD and BIM environments, generated through rule-based or procedural extraction from point clouds and LiDAR, or produced through data-driven methods such as AI segmentation, implicit fields, or hybrid reconstruction.

Aspect	**BIM models (documentation, coordination, asset information)**	**Structural FEM models (Analysis of loads, deformations, stability)**	**Energy models (Thermal and daylight simulation, operational performance)**
Geometry	Explicit solids using B-Rep or constructive solid geometry; parametric building objects with constraints.	Discretized meshes of nodes and elements (bars, beams, shells, solids); mid-surface abstractions common.	Thermal zones as volumes; surfaces and sub-surfaces represented with BRep faces; adjacency graphs between zones and exterior.
Topology	Object hierarchies and assemblies; rooms, spaces and systems linked through relationships.	Element connectivity graphs, degrees of freedom, boundary conditions, constraints and load cases.	Zone-to-surface-to-edge relationships; apertures nested within parent surfaces; view factors or radiosity matrices where relevant.
Semantics	Rich attributes for materials, properties, classifications and relationships; persistent identifiers for change tracking.	Material laws, sections, nonlinearities, solver settings, step definitions and result fields.	Construction layers, thermophysical properties, schedules, systems, weather data and internal gains.
Typical schemas and files	IFC, STEP, proprietary authoring formats.	Mesh formats and solver decks such as MSH, INP, BDF, DAT and related types	IDF, gbXML, Modelica-based libraries, OpenStudio libraries
Integrity needs	Watertight solids for quantities; consistent local and global coordinates; controlled level of detail and information.	Mesh quality metrics, numbering and sparsity patterns, unit consistency, analysis step histories.	Closed and manifold zones; correct surface orientations; consistent adjacencies; well-defined thermal boundaries.

Table 6. Different types of models and different types of data structure.

Resulting datasets are stored in formats suited to their role, including IFC or Revit families for BIM, CityGML or CityJSON for city-scale semantics, and OBJ, PLY or glTF for visualization, and can be converted into computational meshes for energy, structural, or environmental simulation. Together, these representations form a continuum from descriptive geometry to analysis-ready and predictive DTs that integrate geometry, semantics, topology, and physics.

Software platforms such as FME play an important role in this process because they are designed to read, parse, transform and rewrite models across many formats, enabling the consistent exchange and repurposing of data required for different simulation and analysis tasks.

2.11 Required Geometric Representations and Data Structures for DTs Applications

The review of Sections 2.2 to 2.7 shows that DT applications require fundamentally different geometric representations and data structures, depending on the target domain and purpose of the model.

In addition, Table 7 further demonstrates that BIM, structural FEM, and energy models differ fundamentally in their geometry representations, topological structures, semantic content, data schemas, and integrity requirements. As a result, direct model reuse across these domains is inherently infeasible, and the notion of a single unified model capable of satisfying all Digital Twin application requirements is unrealistic.

Energy application models require explicit geometric representations, such as those provided by BIM and other topologically explicit models, volumetric and surface-based models used for simulation purposes; in addition to geometry, such models must also incorporate semantic information, which can be efficiently managed using graph-based data structures, in line with recent research trends.

As illustrated in Figure 11, the practical implementation of DTs demands a multi-level modeling approach capable of coupling heterogeneous geometric representations with diverse data structures. Some geometric models, particularly BIM, are highly structured and topologically explicit but largely static. In contrast, other models are more dynamic, lightweight, and easy to generate, yet they lack the structural and semantic richness required for advanced simulation and data enrichment tasks.

Aspect	BIM Models	Structural FEM Models	Energy Models
Geometry	Explicit solids (B-Rep, CSG)	Discretized meshes	Volumes + B-Rep surfaces
Topology	Object hierarchies	Element connectivity	Zone adjacency graphs
Semantics	Materials, classifications	Material laws, solver settings	Thermophysical properties
Formats	IFC, STEP, RVT	MSH, INP, BDF	IDF, gbXML, OpenStudio
Primary Use	Documentation, coordination	Structural analysis	Energy performance

Table 7. Model Type Requirements for DT Applications.

At present, neither paradigm alone is sufficient to support integrated simulation, real time data coupling, or scalable DT deployment. These limitations highlight the need for a structured process that progressively derives consistent and analysis-ready DT from raw reality capture data.

To address these challenges, a multi-level DT is required as a stable spatial backbone that explicitly encodes geometry, topology and semantics, while enabling systematic transformation into domain specific analytical models and integration with time varying sensor data.

Chapter 3 decomposes these high-level requirements into concrete implementation challenges organized across four categories:

1. *General & Strategic challenges:* fragmented workflows, high implementation costs, digital-physical misalignment, data governance.
2. *Methodological problems:* incomplete data, lack of coherent frameworks, dependence on static BIM intermediaries, non-watertight geometry.
3. *Practical & Operational problems:* multi-step construction complexity, heterogeneous tool integration, computational costs, limited automation.
4. *Interoperability problems:* non-bijective format mappings, geometric validity loss, semantic inconsistencies, weak IoT-BEM coupling.

For each challenge category, Chapter 3 then presents the Scan-to-EDT framework components, in particular (i) strategic, (ii) technical, (iii) interoperability, and (iv) operational solutions, that systematically address the identified gaps and enable robust DT deployment presented in Chapter 4.

3

Chapter 3 bridges the gap between abstract problems and concrete solutions by decomposing the three foundational issues identified in Chapter 1 into actionable challenges and corresponding framework components. The chapter categorizes real-world Digital Twin development obstacles into four groups: strategic challenges, methodological problems, operational issues, and interoperability barriers.

Each category is analyzed to show how issues such as non-watertight geometry, vendor lock-in, and lossy format conversions arise in practice. The chapter then introduces a unified framework composed of nine solution components organized into strategic, technical, interoperability, and operational layers. Each component is explicitly mapped to the problems defined in Chapter 1, ensuring that the proposed framework addresses root causes rather than symptoms and establishing a clear foundation for the implementation and validation presented in subsequent chapters.

Chapter 3

3 Challenges and proposed solutions

As shown in Chapter 2, the development of a building Digital Twin (DT) involves multiple tightly coupled steps, including data acquisition, preprocessing, structuring, 3D modelling, simulation, and analysis. Integrating these steps into a single end-to-end framework remains challenging, as inconsistencies and errors introduced at early stages propagate and intensify throughout the pipeline.
Building on the core problems identified in Chapter 1 (Problems 1–3), this chapter decomposes these high-level issues into the concrete challenges that emerge in real-world DT development. These challenges are grouped into four categories (Fig. 21):

1. *General & strategic challenges*, concerning the overall definition, scope, value, and lifecycle governance of the DT framework.
2. *Methodological problems*, concerning data models, transformation rules, coupling between BIM–BEM–IoT layers and procedures for calibration and validation.
3. *Practical & operational problems*, concerning pipeline orchestration, automation, integration with existing operational technology (OT) systems and computational performance in real deployments.
4. *Interoperability problems*, concerning heterogeneity of exchange formats (e.g. IFC, gbXML, epJSON), non-equivalent semantics and lossy geometric conversions.

The challenge categories defined in this chapter directly correspond to the foundational problems introduced in Chapter 1. Strategic fragmentation and operational inefficiencies relate to Problem 1 (end-to-end integration), methodological limitations affect Problem 2 (model validity and reliability), and interoperability failures directly address Problem 3 (cross-domain semantic consistency).

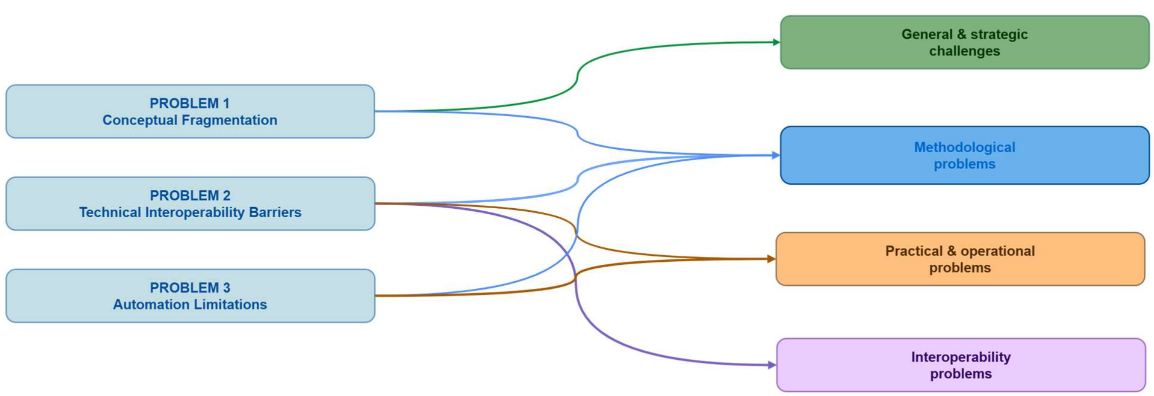


Fig 21. Correlation between the problems identified in Chapter 1 and the challenge categories in Chapter 3.

3.1 General and Strategic Challenges

At the strategic level, DT development is hindered by fragmented processes, high costs, and limited interoperability across tools and domains.

- **Fragmented workflows**

  DT development spans many phases, such as data acquisition, BIM or geometry modelling, semantic enrichment, simulation setup, calibration, monitoring and visualization. These tasks are typically handled by different actors using heterogeneous tools. Handover processes are often based on static file exchanges rather than shared or federated environments, creating siloed and weakly connected workflows.

- **High implementation costs for software and tools**
  Real-world DT projects usually rely on BIM software, GIS platforms, simulation engines, databases, IoT middleware, and visualization environments. Licensing, cloud services and maintenance—combined with the need for specialized expertise—make deployment and scaling economically demanding.

- **Digital–physical misalignment over time**
  Changes in the physical asset—refurbishment, system replacement, occupancy shifts—are rarely captured systematically in the DT. Without robust update procedures and versioning, the digital representation progressively diverges from reality.

- **Data privacy and ownership**
  Sensor-rich environments generate detailed information about occupancy, behaviour, and comfort. Ownership, retention periods, and sharing conditions are often unclear, creating legal and ethical constraints for advanced DT services.

The challenges described above vary in their severity, impact on the DT pipeline and urgency of resolution. Table 8 summarizes these issues by categorizing them according to priority, challenge type, key technical implications and expected urgency for action. These priority levels highlight which challenges pose immediate risks to DT deployment, particularly interoperability and data quality, and which ones primarily affect long-term scalability and operational robustness. The next section examines the methodological problems that arise when these challenges propagate through the modelling and simulation pipeline.
Technical, operational and interoperability-specific issues are detailed in Sections 3.2–3.4.

Priority	Challenge Category	Key Issues	Impact	Section
Critical	Interoperability	Data loss during format conversion (IFC → gbXML → epJSON); inconsistent semantics across BIM, BEM and DT representations; non-bijective mappings	Disrupts the entire DT pipeline and prevents reliable cross-domain integration; invalidates simulation fidelity and traceability	Section 3.4
Critical	Technical Data Quality	Incomplete or noisy input data from sensors and legacy documentation; non-watertight, non-manifold geometry from automated reconstruction	Compromises all downstream modeling, simulation, and analysis tasks; requires extensive manual repair	Section 3.2
High	Workflow Fragmentation	Lack of coherent end-to-end framework; siloed processes across acquisition, modeling, simulation, and operation; static file exchanges	Limits scalability, reuse, and reproducibility across projects; creates weakly connected workflows	Section 3.1 Section 3.2
High	Workflow Automation	Dependence on manual steps for geometry fixing, semantic validation, zone definition and sensor mapping; weak toolchain integration	Reduces efficiency, repeatability, and reliability of Scan-to-EDT processes; increases update time	Section 3.3
Medium	Operational Scalability	Methods designed for single buildings fail at district or city scale due to increased model complexity and data volume	Restricts deployment to small-scale use cases; prevents urban-scale DT adoption	Section 3.1 Section 3.3
Medium	Implementation Cost	High software licensing, cloud infrastructure, and specialized expertise requirements for integrated Scan-to-EDT workflows	Creates economic barriers to adoption, particularly for SMEs and public sector organizations	Section 3.1
Low	Operational Interpretability	Black-box or grey-box DL models with limited transparency; unclear failure modes and validation mechanisms	Raises regulatory, validation, and trust concerns in engineering-grade applications; hinders error diagnosis	Section 3.2

Table 8. Key challenges in DT development and deployment across technical, operational, and strategic dimensions.

3.2 Methodological problems

Methodological issues concern how data, models and representations are structured and transformed into a coherent, simulation-ready and operational DT.

- **Incomplete or noisy input data**
  Real-world input data are often sparse, outdated, or inconsistent. BIM models may lack simulation attributes; sensor data may be faulty or irregular, and legacy documentation may be unreliable.

- **Lack of a coherent end-to-end framework**
  There is no widely adopted methodological framework for DT development in the building and energy domains. Workflows are often project-specific, with unclear boundaries between data acquisition, modelling, simulation, calibration, and operation.

- **Dependence on BIM as an intermediary**
  IFC-based BIM is frequently used as the central intermediary for DT generation, but BIM remains fundamentally static and not optimized for simulation or real-time data integration. IFC geometry is cumbersome to edit, often requires re-export cycles and suffers from inconsistent software implementations.

- **Black- or grey-box models with low interpretability**
  Data-driven and DL components may achieve high predictive accuracy but offer limited interpretability, hindering error diagnosis, transparency, and regulatory compliance.

- **Limited predictive reliability of DL models**
  DL models often generalize poorly to unseen buildings or climates. Performance degrades when input data are sparse, noisy, or outside the training distribution.

- **Non-watertight, non-manifold geometry**
  Automated reconstruction pipelines often yield geometry that violates manifoldness or watertightness, preventing reliable simulation without extensive manual repairs.

- **Regulatory and semantic gaps**
  Simulation-ready models often lack explicit representations of regulatory constraints or operational semantics (e.g. setpoints, schedules, device behaviour), reducing realism and comparability of results.

These issues primarily affect model validity, simulation readiness and the correctness of cross-layer coupling.

3.3 Practical & Operational Problems

Operational issues arise from the need to integrate multiple steps into a coherent, robust DT pipeline.

- **Multi-step construction of a structured model**
  DT construction typically involves: (i) data collection, (ii) cleaning and normalization, (iii) geometry generation, (iv) semantic assignment, (v) zoning and systems definition, (vi) sensor linking and (vii) simulation setup. Errors can propagate across steps, making full automation difficult.

- **Need for integration across heterogeneous tools**
  Many steps are developed independently, requiring DT developers to manually reconstruct workflow logic, resolve inconsistencies and create unified data structures.

- **High computational cost**
  Long-term sensor streams, high-resolution 3D models and multiple simulation scenarios impose significant computational demands, especially during classification, calibration, and optimization.

- **Limited automation**
  Manual interventions remain common geometry fixing, semantic validation, zone definition, sensor mapping, reducing repeatability and increasing update time.

- **Poor scalability to the district level**
  Methods that work for individual buildings often fail at the neighborhood or city scale due to increased model complexity and data volume.

- **Lack of domain-specific datasets**
  Few high-quality, labelled datasets exist for DT tasks, limiting the development and evaluation of data-driven components.

- **Constraints from simulation tools and APIs**
  Simulation engines impose strict requirements on geometry, zoning, schedules and semantics. Adapting DT data to these constraints requires additional conversion layers and custom interfaces.

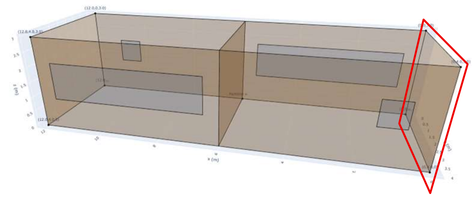
Overall, interoperability and data quality represent immediate critical issues, while automation and fragmentation constrain repeatability and scaling. The remaining issues primarily influence long-term maintainability and trust.

3.4 Interoperability Problems

Interoperability is a cross-cutting issue that affects all stages of the DT pipeline, from model generation and semantic enrichment to simulation and operational integration. In practice, DT workflows rely on multiple domain-specific standards and tools, each adopting different geometric primitives, data structures, and semantic assumptions. As a result, even "standardized" formats remain only partially compatible, and building information is frequently degraded or reinterpreted during exchange.
To illustrate this heterogeneity, Table 9 compares how the same physical wall element is represented across common formats used in the DT pipeline.
The comparison highlights that a single building component is encoded differently depending on the target domain and its modeling requirements:

- *IFC (BIM):* the wall is modeled as an object within a hierarchical project structure:
  Project → Building → Storey → Space → Wall, Floors, Ceilings → Openings.
  Geometry is typically expressed using boundary representation (B-Rep) and placement operators. Semantics (e.g., type, relationships, materials) are encoded through entity references and property sets, while spatial relationships such as containment and adjacency are represented indirectly through the schema structure.
- *FEM input:* the wall is discretized into nodes and finite elements. Geometry is defined by nodal coordinates, while topology and behavior are encoded through element connectivity, material constitutive laws, and boundary conditions. The representation is analysis-oriented and not intended to preserve BIM-style object semantics.

Data structure and format file structure:



One wall for example:

```
{"wall":"South","zone":"LivingRoom","plane":"y=0",
 "v":[[0,0,0],[6,0,0],[6,0,3],[0,0,3]],

"win":{"id":"W1",
"v":[[0.5,0,1.0],[5.5,0,1.0],[5.5,0,2.2],[0.5,0,2.2]]}}
```

BIM - IFC

```
#10= IFCPROJECT('P1',$,'SimpleHouse',$,$,$,$,$);
#20= IFCBUILDING('B1',$,'Building_01',$,$,$,$,$,.ELEMENT.,$,$);
#30= IFCWALLSTANDARDCASE('W1',$,'Wall_LivingRoom',$,$,#101,#501,$);
#31= IFCWALLSTANDARDCASE('W2',$,'Wall_Bedroom',$,$,#102,#502,$);
#40= IFCSPACE('S1',$,'LivingRoom',$,$,#201,$,$,.ELEMENT.,$);
#41= IFCSPACE('S2',$,'Bedroom',$,$,#202,$,$,.ELEMENT.,$);
#101= IFCPOLYLINE((0.,0.,0.),(6.,0.,0.),(6.,4.,0.),(0.,4.,0.),(0.,0.,0.));
#501= IFCPRODUCTDEFINITIONSHAPE($,$,(#101));
#201= IFCLOCALPLACEMENT($,#101);
```

FEM

```
*NODE
1, 0., 0., 0.
2, 6., 0., 0.
3, 6., 4., 0.
4, 0., 4., 0.
5, 0., 0., 3.
6, 6., 0., 3.
7, 6., 4., 3.
8, 0., 4., 3.
*ELEMENT, TYPE=SHELL, ELSET=WALL_LR
1, 1,2,6,5
*MATERIAL, NAME=BRICK
*ELASTIC
2.10E7, 0.20
*SHELL SECTION, ELSET=WALL_LR, MATERIAL=BRICK
0.30
```

Green Building XML (gbXML)

```
<gbXML>
  <Campus>
    <Building id="B1">
      <Space id="LivingRoom"/>
      <Space id="Bedroom"/>
      <Surface id="Wall_LR" surfaceType="ExteriorWall">
        <AdjacentSpaceId spaceIdRef="LivingRoom"/>
        <PlanarGeometry>
          <PolyLoop>
            <CartesianPoint><Coordinate>0</Coordinate><Coordinate>0</Coordinate><Coordinate>0</Coordinate></CartesianPoint>
            <CartesianPoint><Coordinate>6</Coordinate><Coordinate>0</Coordinate><Coordinate>0</Coordinate></CartesianPoint>
            <CartesianPoint><Coordinate>0</Coordinate><Coordinate>0</Coordinate><Coordinate>3</Coordinate></CartesianPoint>
            <CartesianPoint><Coordinate>0</Coordinate><Coordinate>0</Coordinate><Coordinate>3</Coordinate></CartesianPoint>
          </PolyLoop>
        </PlanarGeometry>
      </Surface>
    </Building>
  </Campus>
</gbXML>
```

epJSON

```
{  "Building": { "SimpleHouse": { "north_axis": 0.0 } },
  "Zone": {
    "LivingRoom": {},
    "Bedroom": {}  },
  "BuildingSurface:Detailed": {
    "Wall_LR": {
      "surface_type": "Wall",
      "zone_name": "LivingRoom",
      "construction_name": "BrickWall",
      "vertices": [
        [0,0,0],
        [6,0,0],
        [6,0,3],
        [0,0,3]
      ]    }  },
  "Construction": {
    "BrickWall": { "layers": ["BrickLayer"] }  },
  "Material": {
    "BrickLayer": { "thickness": 0.3, "conductivity": 0.72, "density": 1800 }  }}
```

Table 9. Comparison of how a single wall element is represented across different formats (IFC, FEM, gbXML and epJSON), illustrating the heterogeneity of geometry encodings, semantics, and simulation requirements.

- *gbXML (energy exchange):* the wall is represented as a planar *surface* defined by an ordered set of vertices inside an XML structure. It is associated with thermal zones, constructions, prioritizing heat-transfer modeling over construction detailing. Object identity and higher-level BIM relationships are simplified.
- *epJSON (EnergyPlus):* the wall is expressed as nested key–value objects (e.g., *BuildingSurface:Detailed*, *Construction*, *Material*), storing vertices, construction layer definitions, and material properties in a form aligned with EnergyPlus requirements. The schema is simulation-ready but differs from BIM in both semantics and object granularity.

This comparison illustrates why robust cross-domain interoperability remains difficult: the same physical entity is transformed into representations with fundamentally different objectives, resulting in semantic mismatches and non-equivalent geometry.

Beyond representational differences, several interoperability issues commonly emerge in DT pipelines:

- *Non-bijective mappings and information loss:* transformations between IFC, gbXML and epJSON are rarely one-to-one. Repeated conversions can remove or distort constructions, material layers, system parameters, and device metadata, limiting downstream simulation fidelity and traceability.
- *Loss of geometric validity:* geometry translations often introduce tessellation artifacts, non-manifold edges, or non-watertight surfaces. Such defects can invalidate zoning, adjacency detection, and simulation meshing, and typically require manual repair.
- *Semantic inconsistencies across domains:* object hierarchies (building–storey–space), adjacency relations (space–surface–zone), and dependencies (system–device–circuit) are not preserved consistently across formats. As a result, essential relationships for control, monitoring, and energy modeling must be reconstructed using custom rules or ad hoc scripts.
- *Weak direct link between DT, IoT and BEM layers:* even when sensor data are available, mappings to simulation entities (zones, surfaces, systems, equipment) are often incomplete or unstable. This limits real-time calibration, fault detection, and control-oriented applications.

These issues motivate the need for explicit interoperability mechanisms, such as semantic mediation layers, validation checks, and multi-representation core models, which are addressed in the proposed framework (Section 3.8).

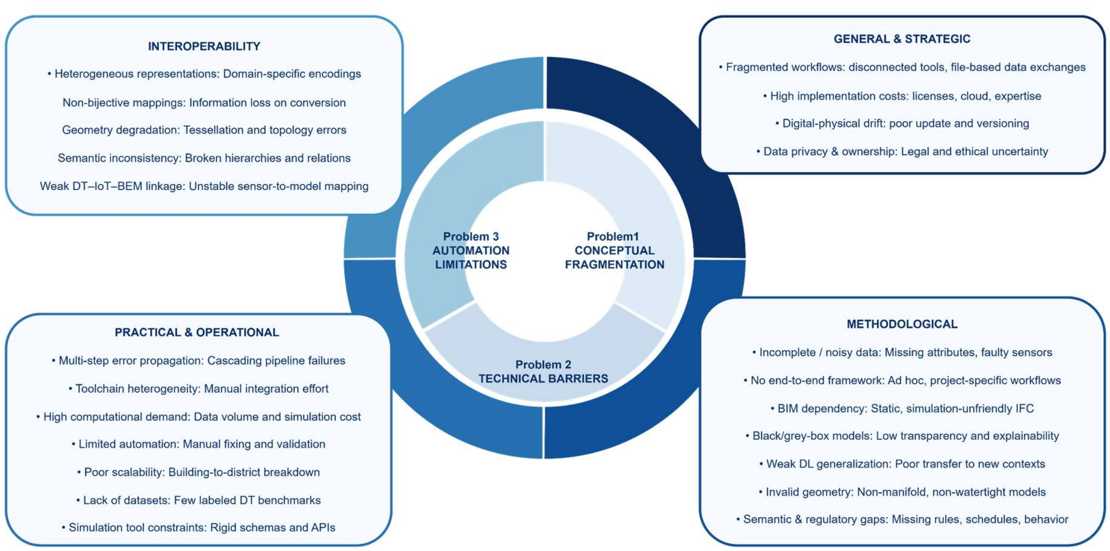


Fig. 22. Problem landscape and correspondence between high-level challenges and detailed technical issues.

Figure 22 summarizes the core problem dimensions introduced in Chapter 1—conceptual fragmentation, technical barriers, and automation limitations—and maps them to the detailed challenges analyzed in Chapter 3. General and strategic issues are expanded into methodological, practical, and operational, and interoperability-specific problems, making explicit how high-level barriers propagate through the Digital Twin pipeline and manifest as concrete technical limitations.

3.5 Proposed solutions & Framework

The proposed framework addresses the challenges in Sections 3.1–3.4 by separating (i) *internal consistency* from (ii) *external exchange*.
Internally, the DT is managed through a multi-representation core model that preserves geometry, topology, semantics, and traceability. Externally, dedicated transformation and validation layers control export to domain formats (e.g., IFC, gbXML, epJSON) and quantify information loss. Finally, the automated framework integrates all steps into a reproducible pipeline suitable for iterative updates related to simulation outputs during operation.

The framework is structured around a multi-representation (or multi-level) core model, explicit transformation and validation layers and an automated execution architecture, which are detailed in Sections 3.6–3.9. Unlike BIM-centric pipelines, the proposed framework does not rely on a single intermediary representation but maintains multiple representations to support validation, automation, and simulations.
Table 10 summarizes how each framework component addresses the challenge categories.

3.5.1 Strategic solutions

Strategic solutions address how a Digital Twin is defined, governed, and maintained over its lifecycle. As summarized in Table 10, these solutions respond directly to fragmented workflows, high implementation costs, and the long-term misalignment between digital models and physical buildings.

	Solution Component	Problem Addressed (Chapter 1)	Main Issue	What It Enables
1	**Unified Digital Twin Framework**	**Problem 1:** Lack of end-to-end integration	Fragmented and disconnected workflows	A single, coherent process from data acquisition to operation
2	**Open Source Environment**	**Problem 1:** Limited scalability and reuse	High software costs and vendor lock-in	Lower costs and community driven, reusable development
3	**Data Quality Checking**	**Problem 2:** Low model validity and reliability	Noisy, incomplete, or inconsistent input data	Simulation ready and reliable models
4	**Multi Representations Core Model**	**Problem 3:** Cross domain inconsistency	Incompatible BIM, FEM, and energy representations	Consistent geometry and semantics across domains
5	**Modular Workflow**	**Problem 1:** Manual and error prone processes	Low automation and poor repeatability	Automated, scalable, and reproducible workflows
6	**Semantic Layer**	**Problem 3:** Semantic inconsistency	Loss or change of meaning during format conversion	Preservation of meaning across tools and formats
7	**Conversion Checks**	**Problem 3:** Information loss during exchange	Undetected geometry and semantic errors	Early detection and reduction of conversion errors
8	**Rules and Regulation Knowledge Base**	**Problem 2:** Missing semantic and regulatory knowledge	Lack of explicit regulatory constraints	Automated compliance checking and realistic simulations
9	**Multi-Level of Detail**	**Problem 1 and Problem 2:** Poor scalability	Fixed model resolution	Adaptive detail from single buildings to districts

Table 10. Practical solution components addressing common technical and organizational challenges in DT workflows.

The *Unified Digital Twin Framework* (Table 10, 1) provides a single end-to-end workflow that connects data acquisition, modeling, simulation, and operation. By defining clear inputs, outputs, and validation steps for each stage, it reduces fragmentation and clarifies responsibilities across tools and stakeholders.

The *Open-Source Environment* (Table 10, 2) addresses high software and licensing costs by providing an extensible core for preprocessing, conversion, and validation. This reduces vendor lock-in and supports transparency, reuse, and community driven development.

Together, these components establish a stable governance structure that supports continuous DT operation rather than one off project delivery.

3.5.2 Technical solutions

Technical solutions focus on data quality, model reliability, and automation within the DT pipeline.
As shown in Table 10, these components directly address noisy data, incompatible representations and error-prone manual workflows.
The *Data Quality Checking Pipeline* (Table 10, 3) automatically cleans, validates, and repairs input data before simulation. This includes filtering sensor noise, fixing non-watertight geometry and verifying basic consistency rules to ensure simulation readiness.
The *Shared Core Model* (Table 10, 4) provides a common internal representation that preserves geometry and semantics across BIM, energy and structural models. By separating internal consistency from external exchange, it avoids repeated information loss during conversion.
Automation is enabled through the *Modular Workflow* (Table 10, 5), which orchestrates conversion, validation, and simulation tasks. This reduces manual intervention, improves repeatability and supports scalable deployments.

These solutions improve reliability while reducing the effort required to build and update DTs.

3.5.3 Interoperability solutions

Interoperability solutions ensure that information retains its meaning when exchanged between tools and formats. As highlighted in Table 10, these components directly address semantic inconsistency and systematic data loss.
The *Semantic Mapping Layer* (Table 10, 6) uses explicit mappings to ensure that objects, relationships, and attributes are interpreted consistently across BIM, energy, and simulation formats.
To detect and reduce information loss, *Conversion Quality Checks* (Table 10, 7) perform automated validation and round-trip testing between formats such as IFC, gbXML and epJSON.
Regulatory constraints are handled through the *Rules and Regulation Database* (Table 10, 8), which encodes standards and codes to support automated compliance checking and realistic simulations.

3.5.4 Operational solutions

Operational solutions address scalability and performance during real-world deployment.
As shown in Table 10, these solutions enable the framework to function from single buildings to district-scale applications.
The *Level of Detail* component (Table 10, 9) dynamically adapts model resolution to balance accuracy and computational cost. This allows high detail where needed for simulation or analysis while maintaining efficiency at larger spatial scales.

3.6 Summary

Together, the strategic, technical, interoperability and operational components summarized in Table 10 form a coherent and integrated DT framework.
Each component directly addresses one or more challenge categories identified in Sections 3.1–3.4, ensuring traceable links between problems and solutions.

By separating internal consistency from external exchange, enforcing data quality and validation, and enabling automation and scalability, the proposed framework overcomes key limitations of BIM-centric and ad hoc DT pipelines. The following chapter demonstrates the practical implementation of these components and evaluates their effectiveness through case studies and validation experiments.

The Scan-to-EDT framework is primarily intended for high-value application domains where topological and semantic rigor directly conditions the quality of downstream analysis. These include: (i) deep energy renovation of existing complex buildings, (ii) energy certification of heritage assets, (iiii) BIM-enabled facility management requiring continuous model updates, and (iv) Digital-Twin-driven building performance monitoring and control. For simpler assessments where approximation and manual abstraction are acceptable, lighter workflows may be more appropriate — the framework is not intended to replace those practices, but to address the specific class of problems where interoperability, consistency, and simulation fidelity are critical requirements.

4

Chapter 4 presents the implementation and methodology of the proposed Scan-to-EDT framework through a series of experimental results. The chapter demonstrates how unstructured 3D reality-based data, semantic enrichment, and topology-aware modeling are integrated into a coherent Digital Twin pipeline supporting interoperability, energy simulation, and data-driven analysis.
The workflow is illustrated from the reconstruction of geometric and topological representations to the generation of multi-level Digital Twin models, including volume-based, surface-based, and graph-based abstractions. Particular attention is given to the enrichment of the topological model with semantic attributes, material properties and energy-related parameters, enabling consistent export to multiple domain-specific schemas, such as IFC, gbXML, epJSON and CityGML.
The chapter further evaluates the robustness of the framework through visualization, simulation and monitoring tasks, including sensor integration, thermal field analyes and scenario-based energy assessment. User-driven retrofit scenarios are explored to highlight the flexibility of graph-based representation and its ability to support on-the-fly modifications and comparative analysis.

Chapter 4

4. The Proposed Method: the Scan-to-EDTs framework

This chapter presents the proposed Scan-to-EDTs framework, an end-to-end pipeline that automatically generates Energy Digital Twins (EDTs) directly from unstructured 3D point clouds. Building on the methods introduced in the previous chapters, the framework combines DL-based classification, Scan-to-BIM reconstruction, topologic modeling, and energy-oriented enrichment into a single workflow. The result is a multi-level geometric DT that is watertight, topologically consistent, and semantically structured for energy simulation and device-to-space mapping.

In the following sections, the framework is applied to several real-world case studies, and its performance is evaluated in terms of geometric accuracy, robustness and suitability for energy analysis and DT applications (Fig. 23).

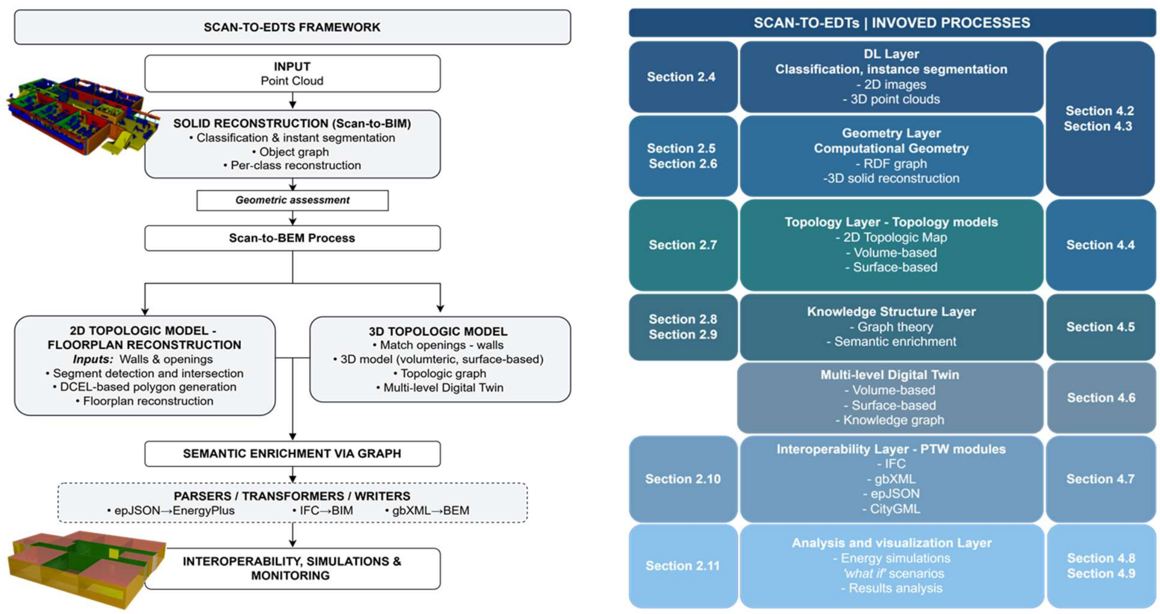


Fig. 23. Overview of the proposed Scan-to-EDTs framework and correspondence between methodological components.

To address the issues identified in Chapter 3, the proposed framework is designed around four core principles.

First, an open-source workflow is adopted, with all major components built on widely used libraries (e.g., *Point Transformer* (Wu et al., 2024), *Geomapi* (Bassier et al., 2024), *TopologicPy* (https://topologic.app/topologicpy_doc/), *Shapely* (Gillies et al., 2025), *IfcOpenShell* (https://docs.ifcopenshell.org/index.html), and *EnergyPlus* (Naeem et al., 2025)), eliminating licensing constraints while remaining suitable for offline and historical data processing, despite current limitations in cloud-native and real-time deployment.

Second, a data-centric processing strategy is employed, where file formats are treated as output rather than primary working representations; internal operations rely on lightweight tabular structures (pandas *DataFrames*) and simple text-based formats, with IFC, gbXML and epJSON generated only at the export stage to reduce coupling to specific schemas or authoring tools.

Third, minimal data loss is ensured through a single unified internal data model that underpins the entire pipeline; parser–transformer–writer (PTW) modules handle conversions to target formats while preserving geometric, semantic, and topological consistency, with quantitative evidence of reduced information loss reported in the Results chapter. Finally, automation and reduced manual effort are emphasized, with geometry reconstruction, topology generation, semantic enrichment, and export largely automated, limiting human intervention to configuration and quality control and enabling rapid regeneration of models when input data change.

Under these principles, the Scan-to-EDTs framework supports multiple Digital Twin operations on a shared underlying model, including the analysis of historical performance data, the generation and updating of simulation-ready geometries, and the progressive enrichment of the DT through graph-based integration with additional datasets.

4.1 The prosed method: overview

The Scan-to-EDTS framework, implemented in modular form as shown in *Appendix A*, addresses the limitations identified in Chapter 3 by providing an end-to-end, open-source pipeline that transforms raw point clouds into simulation-ready Energy Digital Twin models. It is organized into seven interconnected phases:

1. **Scan-to-BIM process (Section 4.2)**
   The proposed Scan-to-BIM workflow integrates deep learning–based classification and instance segmentation of point clouds using Point Transformer v3, YOLOv8, and GroundingDINO. The semantically enriched point cloud is then structured into an RDF-based property graph in Geomapi Python-based library[1], which serves as the foundation for per-class, rule-driven 3D parametric reconstruction of building elements.
2. **Scan-to-BEM process (Section 4.3)**
   Development of an in-house visualization interface that allows users to navigate the point cloud and select classes to interactively pick points corresponding to devices and energy-related elements. This tool enables the extraction and refinement of BEM-relevant classes and device positions needed for downstream energy modelling.
3. **Topologic model creation (Section 4.4)**
   Starting from the classified and reconstructed objects, volumetric and surface-based representations are generated, with explicit spaces, boundaries and adjacencies. The 2D Topologic Map is used to derive a fully 3D, topologically consistent model.
4. **Topologic Graph (Section 4.5)**
   A graph-based structure combines and organizes all topologic information, embedding the attributes of the volumetric and surface models. This graph enables the construction of the Multi-Level DT and forms the backbone of the geometric DT (gDT), ready to be enriched with IoT and other data streams.
5. **Multi-Level DT (Section 4.6)**
   Integration of the volumetric model, the surface-based B-Rep model and the topologic graph into a unified multi-level representation. This composite structure constitutes the core output of the proposed framework: a geometric DT enriched with semantics, hierarchy, and topological relations.
6. **Leveraging the graph structure (Section 4.7)**

[1] https://ku-leuven-geomatics.github.io/geomapi/

Assignment of materials, constructions and devices to the topologically organized geometry, producing a domain-ready model for energy and control applications. The same graph structure enables the export of the internal model to standard AEC formats (IFC, gbXML, epJSON) for interoperability with BIM, BEM, and DT tools.

7. **Energy simulation (Section 4.8 – Section 4.9)**
   Integration with EnergyPlus and CFD tools to perform energy simulations, calibration, and scenario testing, enabling performance analysis and validation of the resulting DT.

4.2 Scan-to-BIM process

The Scan-to-BIM methodology adopted within the Scan-to-EDT framework is detailed in (Roman et al., 2024b). It implements a fully automated pipeline that converts unstructured 3D reality-capture point clouds into structured, parametric, and IFC-compliant BIM representations. Following point cloud subsampling and spatial partitioning, deep learning–based semantic and instance segmentation is applied, and detected building elements are encoded as nodes in a semantic graph enriched with geometric and topological metadata. Within Scan-to-EDT, the core contribution lies in the 3D reconstruction stage, which transforms segmented point clusters into parametric geometries using computational geometry and topology-aware modeling. Geometric primitives are estimated via RANSAC-based plane fitting, orientation analysis, Gaussian refinement, and proximity-driven clustering. Walls are reconstructed by deriving centerlines, orientations, thicknesses, and vertical extents from floor and ceiling reference levels, while topological consistency is enforced through intersection analysis and neighbor validation. Columns are reconstructed using refined minimum oriented bounding boxes based on convex hull projections and statistical filtering, and openings are identified as geometric discontinuities within wall surfaces subject to dimensional and positional constraints.

Although the full Scan-to-BIM workflow is described in subsequent sections, within the Scan-to-EDT framework the process terminates at the generation of coherent, structured, and topologically consistent meshes, which serve as input to the Topologic model definition pipeline.

4.2.1 Classification: working on point clouds

The point cloud–based classification workflow leverages the Pointcept[2] framework, which integrates PointTransformer v3 (PTv3) for semantic classification and the *Segmentor* module for instance segmentation, ensuring reliable identification of building elements.

The pipeline begins with preprocessing to improve geometric consistency and reduce computational load: point clouds are subsampled using voxel-grid filtering to a uniform spacing of 1 cm, normals are estimated via PCA on local neighborhoods of 20–40 points to stabilize surface orientation and the data is partitioned into grid-based tiles of 5 m to enable batch processing within GPU memory constraints.

The model is trained to detect six indoor AEC classes, covering primary structural elements (walls, floors, ceilings, and columns) and secondary elements (doors, windows). Each tile is fed to the PTv3-based semantic segmentation network in *Pointcept*, with the *Segmentor* configured with a multi-scale receptive field (radii of 0.4 m, 0.8 m and 1.2 m). The network outputs per-point logits, which are converted into final semantic labels by applying a `softmax` confidence threshold in the range 0.5–0.7.

As reported in (Roman et al., 2024), the detection stage is particularly robust for objects with strong geometric signatures, such as floors and walls, where normals allow a strong plane definition and consequent

[2] https://github.com/Pointcept/Pointcept

element extraction. Following classification, points of the same class are clustered using connected components or DBSCAN (ε = 5–10 cm, *minPts* = 150) to extract coherent object instances. For each cluster, geometric primitives are computed, including oriented bounding boxes via PCA, plane fitting via RANSAC (distance threshold 1–3 cm, maximum 2000 iterations), centroid and axis estimation.

Figure 24 illustrates the current limitations of small-object detection, using columns as an example, while Figure 25 presents the results of instance segmentation applied to these columns. State-of-the-art DL methods still struggle to accurately detect small objects. In this work, this issue is addressed by:

- operating on rasterized images derived from the point cloud to detect openings (columns, windows and doors), and
- manually labelling the position and class of devices such as lights, radiators, and HVAC terminals.

Columns are detected using YOLOv8 (Varghese and Sambath, 2024) and openings using GroundingDINO (Liu et al., 2023), leveraging pre-trained models available online for these classes.

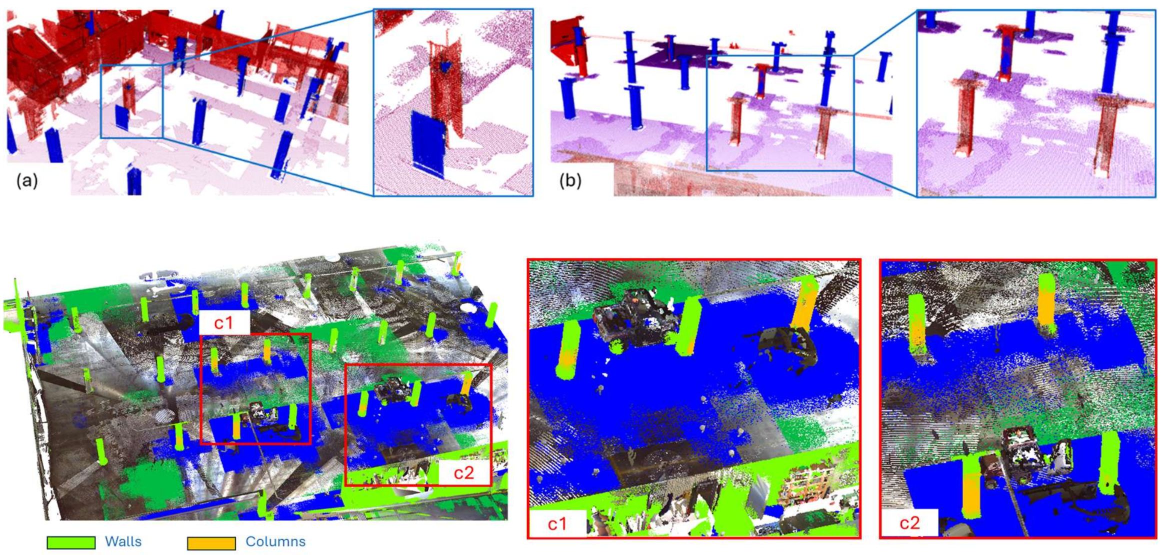


Fig. 24. Example of Deep Learning limitations in object detection, where columns are incorrectly classified as walls in (a, b, c1-c2).

These clusters are assembled into an object RDF graph in which nodes store attributes such as semantic label, bounding-box parameters, normal consistency and local confidence scores, while edges encode spatial relationships (adjacency, intersection, parallelism and coplanarity, assessed using angle thresholds <5–10° and distance gaps < 5 cm).

This enriched geometric–topological structure provides a robust basis for downstream reconstruction, enabling reliable inference of wall boundaries, openings and volumetric envelopes, even in the presence of occlusions, missing regions, or noisy TLS measurements.

4.2.2 Classification: working on images

To address the limitations of purely point cloud–based detection, especially for small, occluded or under-represented classes, the pipeline integrates complementary image-based strategies for columns on one side and openings (doors and windows) on the other.

For columns, the raw point cloud is rasterized into a 1 cm–resolution from top-view projection and aligned with the global *xy* axes using the principal building orientation derived from the global oriented bounding box (OBB). A YOLOv8 detector is applied to this raster image to localize column candidates.

Let $\mathcal{L}_g$ denote column clusters inferred from geometric features, $\mathcal{L}_v$ those obtained from the visual (YOLO-based) detector and G the structural grid estimated by the visual network; the final column set is defined as $\mathcal{L}_f = \mathcal{L}_g + (\mathcal{L}_v \cap G)$.
These candidates are then back-projected into 3D and validated using detection scores, grid compliance, and point cloud geometric signatures (Figure 26). For openings (doors and windows), virtual cameras are placed 1 m orthogonally to each reconstructed wall face, and high-resolution orthographic views are rasterized at 1 cm using wall-aligned octree meshes that combine wall meshes and wall point clouds.

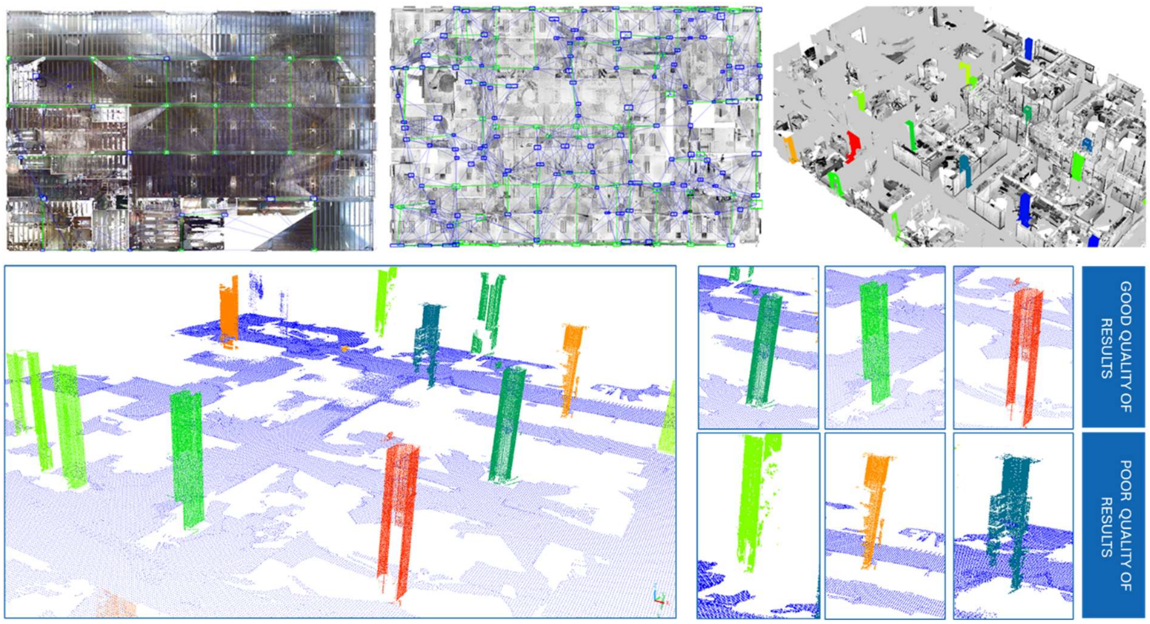


Fig. 25. Vision-based column detection pipeline. Top: Rasterized point clouds are arranged into a structured grid, then processed with YOLOv8 to infer and classify column instances. Bottom: Column detections under partial occlusions and missed/ambiguous regions.

GroundingDINO is applied to these projections to predict oriented bounding boxes for openings, which are filtered using dimensional constraints, where valid doors must satisfy $0.50\,\mathrm{m} \leq w \leq 3.00\,\mathrm{m}$ and $1.50\,\mathrm{m} \leq h \leq 2.30\,\mathrm{m}$, and a topological constraint on the reference floor $\mathcal{F}_{\mathrm{ref}} < 0.50\,\mathrm{m}$, ensuring that detected openings lie close to the floor. All accepted detections (columns and openings) are finally mapped back into 3D as oriented bounding boxes via inverse projection transforms, providing precise object instances for subsequent fusion with point-based predictions and for downstream reconstruction.

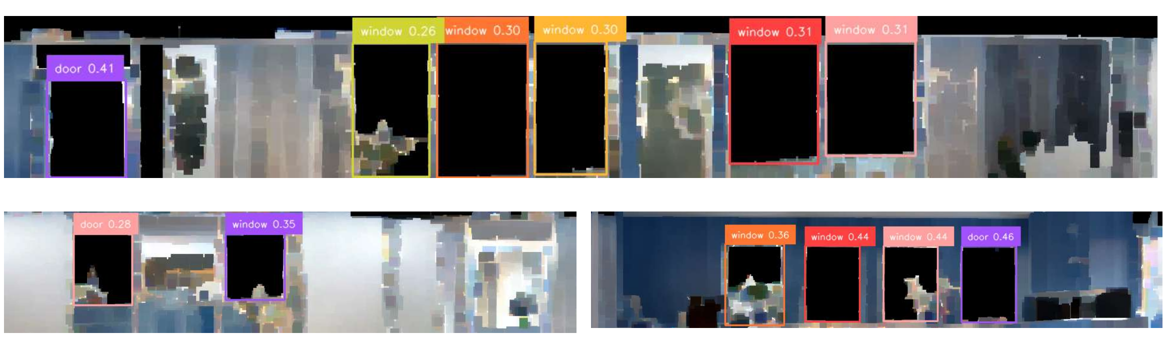


Fig. 26. Zero-shot vision-based detection of openings using Grounding DINO, applied to rasterized representations of point cloud data.

4.2.3 Late fusion approach

To combine the strengths of point-based and image-based predictions, a class-wise late-fusion strategy is adopted. Late fusion (Section 2.4.4) is performed by first running inference independently on the point cloud and image data and then combining their predictions via per-class model selection. Let $p^{(PT)}(x) \in \mathbb{R}^C$denote the per-class logits or probabilities produced by the PointTransformer on the 3D point cloud, and let $p^{(IMG)}(x)$and $p^{(G)}(x) \in \mathbb{R}^C$be those obtained by reprojecting image-based detections from YOLO (for columns) and GroundingDINO (for openings), respectively, onto the point cloud. On a held-out validation set, per-class performance (mIoU) is measured and, for each class c, a provider $\pi_c \in \{PT, IMG, G\}$is selected as (Eq. 1):

$$\pi_c = \arg \max_{k\in\{PT,IMG,G\}} \mathrm{mIoU}_{k,c}. \quad \text{(Eq. 1)}$$

At inference time, the fused score for class c at point x is (Eq. 2):

$$\tilde{p}_c(x) = p_c^{(\pi_c)}(x), \quad \text{(Eq. 2)}$$

and the final predicted label is (Eq. 3):

$$\hat{y}(x) = \arg \max_{c\in\{1,\dots,C\}} \tilde{p}_c(x). \quad \text{(Eq. 3)}$$

Image-based predictions are reprojected into the 3D frame to ensure spatial alignment, after which connected-component smoothing and local majority voting enforce spatial consistency. In practice, geometry-dominant classes (e.g. walls, floors) with clear geometric features are typically taken from the Point Transformer, whereas small or texture or appearance-dominant classes (columns, doors, windows) are sourced from YOLO or GroundingDINO, thereby exploiting the benefits of both modalities without heuristic per-point weighting.

4.2.4 From point cloud to RDF graph

The graph generation (Section 2.5) is implemented using the Geomapi[3] library, specifically leveraging the *PointCloudNode* utilities from the library's *nodes* module. In general, a *Resource Description Framework* (RDF) graph is a is a triple-based graph consisting of *subject*, *predicate*, and *object* relationships.

Schema for RDF graph

```
@prefix e57: <http://libe57.org#> .
@prefix v4d: <https://w3id.org/v4d/core#> .
@prefix xsd: <http://www.w3.org/2001/XMLSchema#> .
```

Wall example

```
<file://first_floor_Walls_15> a v4d:PointCloudNode ;
    e57:cartesianBounds "[-60.7145976 -57.0355976 13.28469111 13.66469111 3.052 6.657 ]" ;
    e57:cartesianTransform """[[ 1. 0. 0. 13.5967254]
                              [ 0. 1. 0. 4.87931762]
                              [ 0. 0. 1. -58.54915832]
                              [ 0. 0. 0. 1. ]]""" ;
    e57:e57Index 0 ;
    e57:pointCount 26355 ;
    v4d:class_id 2 ;
    v4d:class_name "walls" ;
    v4d:color "[0.25098359 0.40784314 0.13333334]" ;
```

[3] https://ku-leuven-geomatics.github.io/geomapi/

```
    v4d:derived_from "first_floor" ;
    v4d:name "first_floor_Walls_15" ;
    v4d:object_id 15 ;
    v4d:orientedBounds "[-58.189066 13.67532339 1.4580346]
                        [-55.43725198 13.74249888 3.42887741]
                        [-59.06239149 13.32100523 6.22091222]
                        [-56.88963194 13.34120229 5.34288262]
                        [-58.63752139 13.36903364 5.81451826]
                        [-56.44660831 13.26979178 5.94088414]
                        [-59.65933814 13.64176018 8.22417459]" .
```

Window example

```
<file://first_floor_Windows_8> a v4d:PointCloudNode ;
    e57:cartesianBounds "[-53.5115976 -46.6495976 29.33369111 30.07569111 4.65791 5.791 ]" ;
    e57:cartesianTransform """[[ 1. 0. 0. 29.78878543]
                                [ 0. 1. 0. 5.3085368]
                                [ 0. 0. 1. -50.69866629]
                                [ 0. 0. 0. 1. ]]""" ;
    e57:e57Index 0 ;
    e57:pointCount 25560 ;
    v4d:class_id 5 ;
    v4d:class_name "windows" ;
    v4d:color "[0.25490196 0.17254902 0.4]" ;
    v4d:derived_from "first_floor" ;
    v4d:name "first_floor_Windows_8" ;
    v4d:object_id 8 ;
    v4d:orientedBounds "[-53.50720774 29.99698789 4.18333599]
                        [-46.58991198 30.27535177 4.62068271]
                        [-53.46260269 29.34912194 4.72038131]
                        [-46.68298469 29.43209191 4.95312815]
                        [-53.60068123 29.36488443 5.78927481]
                        [-46.56861294 29.36488214 6.00788152]
                        [-53.46665831 30.26467468 6.22352707]" .
```

Table 11. Example of structured RDF graph for classified objects in the point cloud.

The result, in this case, is generated directly from the classified point cloud is a typed RDF property graph, where:

- nodes represent building elements and resources such as *Wall, Opening, Mesh and PointSet*;
- edges capture relations between objects such as *boundedBy, derivedFrom and inStorey*.

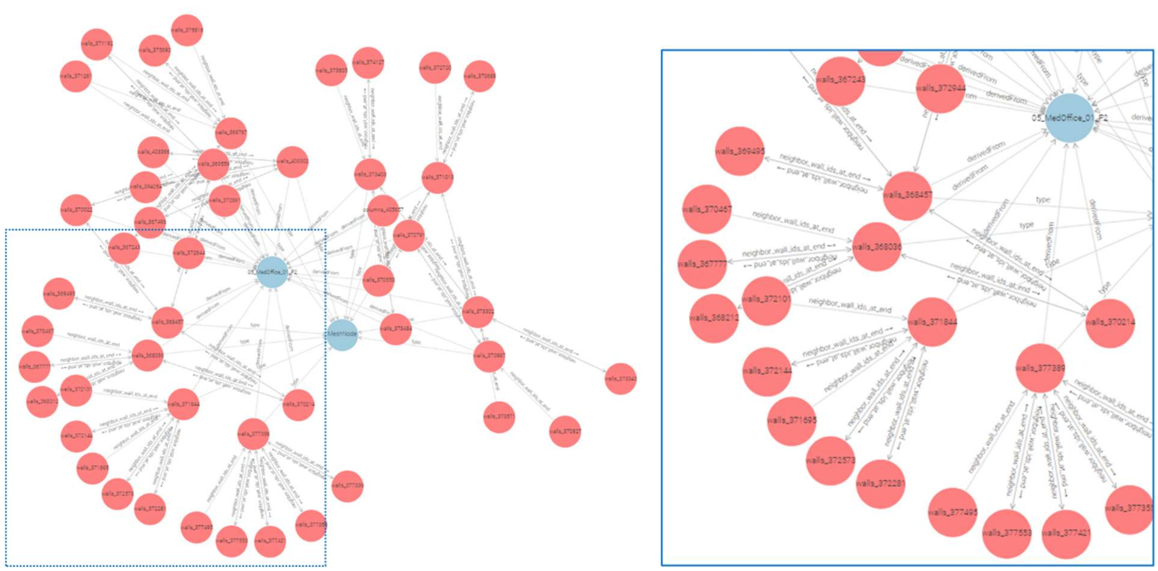


Fig. 27. RDF knowledge graph representation of reconstructed building elements and their semantic and topological relationships. The right panel shows a zoomed-in view highlighting wall entities and neighborhood relations.

Graph (Fig. 27) provides direct access, via *class_id* and *object_id*, to all element-specific information, including rotation matrices, bounding boxes and the sets of points classified for each class and object, which are subsequently used to derive the solid model. This structure supports class-specific reconstruction, where each class is processed by a dedicated modular pipeline, as described in Section 3.4.

Each node stores both semantic and geometric attributes, ensuring complete traceability from the original point cloud segments to the final mesh representations and enabling reproducible, fine-grained reconstruction workflows.

4.2.5 Per-class 3D solid reconstruction

In the downstream Scan-to-BIM stage, computational geometry is applied per class. For each semantic class (Section 2.6), we operate on the corresponding point sets, retrieved from the *points* section of the RDF graph, to manipulate points, fit or extract primitives (e.g., walls), and refine geometry under a Gaussian noise model (e.g., columns). Per-class reconstruction is a standard (Blaha et al., 2016): a class-conditioned pipeline maps each class to a dedicated operator chain. All elements in a class follow the same workflow, irrespective of scale or geometric complexity, with parameters estimated from each element's measurements. The classifier outputs a class label and an object-level scalar field; at run time, the label selects the computation routine, which is executed per element identifier using its associated data.

Let $\mathcal{D}$ represent the complete point cloud scene. After performing the classification phase, the point cloud $\mathcal{D}$ s defined as a finite set of three-dimensional samples endowed with spatial and semantic attributes, that is (Eq. 4):

$$\mathcal{D} = \mathcal{U} + \mathcal{F} + \mathcal{C} + \mathcal{W} + \mathcal{L} + \mathcal{O} \qquad \text{(Eq. 4)}$$

Where:

- $\mathcal{F}$ denotes the floor class, the reference base to which the bases of all walls are anchored,
- $\mathcal{C}$ denotes the ceiling class, the reference plane to which the tops of all walls are anchored,
- $\mathcal{W}$ denotes the wall class, comprising all identified vertical partition elements,
- $\mathcal{L}$ denotes the class columns, comprising all structural columns,
- $\mathcal{O}$ denotes the openings class, collecting all doors and windows.
- $\mathcal{U}$ denotes the unclassified class, containing all elements not included in the sets above.

Next sections will explain for each class the adopted procedure.

4.2.5.1 Floors

Floors are modeled as the base surfaces of future room cells, that is, the bottom faces of the building shells, assumed planar and horizontal. In Scan-to-BIM the key attribute is the elevation z, which anchors the bases of walls. If the floor elevations satisfy $\Delta z = z_{\max} - z_{\min} < 0.25$ m (typical minimum slab thickness), a single representative elevation is applied (Eq. 5):

$$z_{floor,\ mean} = z_{min} + \frac{z_{max} - z_{min}}{2} \qquad \text{(Eq. 5)}$$

Otherwise, floor points are segmented by density-based clustering. Surface normals are first estimated using local Principal Component Analysis (PCA) with a neighborhood radius $r = 0.30$ m, with particular attention to the vertical component n_z. After normal estimation, DBSCAN is applied with a spatial radius of 0.25 mto cluster points that belong to the same plane. Smaller PCA radii from 0.10 to 0.15 m captures finer details but are more sensitive to noise, whereas larger radii from 0.50 to 0.60 m yield smoother slab normals but risk over smoothing and merging adjacent levels. The resulting set of floor elevations $z_i$is stored and during wall generation the base elevation is assigned through a nearest neighbour query between each wall footprint and the detected floor elevation points. The convention of z_{mean} supports the anchoring of wall bases and tops, while for room volumes and the associated surface-based walls, the conventions of finished floor and finished ceiling are adopted, as detailed in Section 4.5.

4.2.5.2 Ceilings

Ceilings are modeled as the top surfaces of room cells, that is, the upper faces of the building shells, assumed planar and horizontal. The key attribute is again the elevation z, which anchors the tops of walls. If the ceiling elevations satisfy $\Delta z = z_{\max} - z_{\min} < 0.25$ ma single representative elevation is used (Eq. 6):

$$z_{ceil,\ mean} = z_{ceil,\ min} + \frac{z_{ceil,\ max} - z_{ceil,\ min}}{2} \tag{Eq. 6}$$

Otherwise, ceiling points are segmented by density-based clustering. Surface normals are estimated with local Principal Component Analysis using a neighborhood radius $r = 0.30$ m with attention to n_z, followed by DBSCAN (Ester et al., 1996) with spatial radius 0.25 m to group points that belong to the same plane. The same radius trade applies as for floors. The resulting set of ceiling elevations $z_{i,ceil}$ is stored, and during wall generation the top elevation is assigned through a nearest neighbour query between the wall footprint and the detected ceiling elevation points. For room volumes and surface-based walls, the convention of finished floor and finished ceiling is used, consistent with Section 4.5.

4.2.5.3 Walls

The topology of walls $\mathcal{W}$ (Fig. 28) is established through a robust workflow based on the methodology in (Bassier et al., 2019). The process begins with noise filtering followed by a Random Sample Consensus (RANSAC)-based plane segmentation to identify dominant supporting planes for each wall cluster $\mathcal{W} = \{w_i, \ldots, w_k\}$.

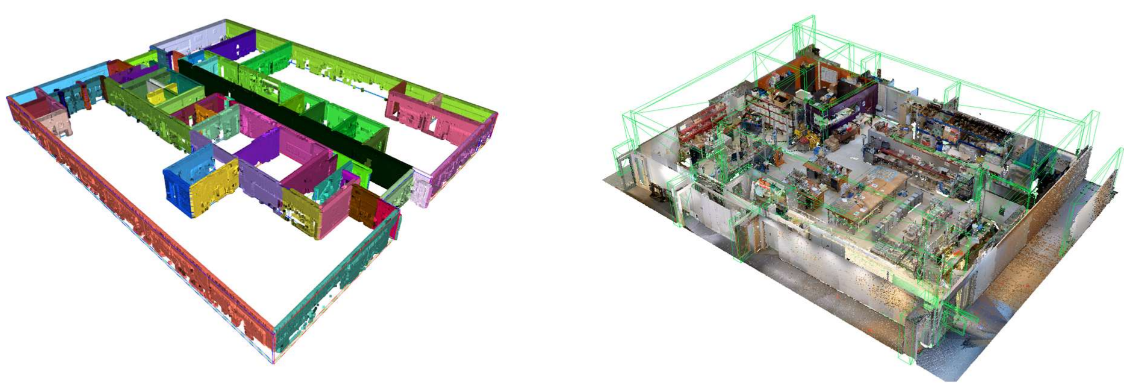


Fig. 28. 3D building reconstruction results. Left: solid mesh model with room-level segmentation. Right: overlay of the reconstructed geometry onto the original point cloud for visual validation.

Each wall plane is expressed in standard form (Equation 7):

$$ax + by + cz + d = 0 \tag{Eq. 7}$$

where a, b, c are the components of the plane's unit normal vector and $d$is the signed distance from the origin. The coefficients are obtained by minimizing the least-squares error (Equation 8):

$$E = \sum_{i=1}^{N} \frac{(ax_i + by_i + cz_i + d)^2}{a^2 + b^2 + c^2} \quad \text{(Eq. 8)}$$

corresponding to minimizing the sum of squared orthogonal distances from the point cloud to the plane. Walls with thickness below a threshold $t_{th} = 0.12$ m are discarded, consistent with the typical minimum size of interior partitions. The remaining walls are grouped into clusters C_i using two criteria: proximity and orientation similarity.

Proximity between clusters $C_1$and C_2 is evaluated using the minimum point-to-point distance (Eq. 9):

$$d(C_1, C_2) = \min_{w_i \in C_1, w_j \in C_2} \parallel w_i - w_j \parallel \leq t_d \quad \text{(Eq. 9)}$$

while orientation similarity is assessed by computing the angle between their normals (Eq. 10):

$$\theta(C_1, C_2) = \arccos\left(\frac{\mathrm{n}_{C_1} \cdot \mathrm{n}_{C_2}}{\parallel \mathrm{n}_{C_1} \parallel \ \parallel \mathrm{n}_{C_2} \parallel}\right) \leq t_o \quad \text{(Eq. 10)}$$

Clusters are merged only when both thresholds are simultaneously satisfied.

To analyze wall connectivity, each wall is represented by its start and end points ($p_{\mathrm{start}}, p_{\mathrm{end}}$), normal vector n, thickness t_i and boundary offsets. The orthogonal boundary points are computed as (Eq. 11):

$$p_{\mathrm{start}}^{\perp} = p_{\mathrm{start}} + \frac{t_i}{2}\mathrm{n}, p_{\mathrm{end}}^{\perp} = p_{\mathrm{end}} + \frac{t_i}{2}\mathrm{n} \quad \text{(Eq.11)}$$

A nearest-neighbour (NN) search identifies potential wall connections by selecting candidates within the distance threshold t_d. Intersections between two wall segments are calculated by solving their parametric line equations (Eqs. 12a–12b):

$$\mathrm{r}_1(t) = p_{\mathrm{start},1} + t(p_{\mathrm{end},1} - p_{\mathrm{start},1}) \quad \text{(Eq. 12a)}$$
$$\mathrm{r}_2(s) = p_{\mathrm{start},2} + s(p_{\mathrm{end},2} - p_{\mathrm{start},2}) \quad \text{(Eq. 12b)}$$

The intersection satisfies (Eq. 13):

$$p_{\mathrm{start},1} + t(p_{\mathrm{end},1} - p_{\mathrm{start},1}) = p_{\mathrm{start},2} + s(p_{\mathrm{end},2} - p_{\mathrm{start},2}) \quad \text{(Eq. 13)}$$

A valid intersection requires $0 \leq t, s \leq 1$. Additional orthogonal checks on boundary offsets ensure that false intersections or wall extensions are removed based on a threshold t_{int}.

To validate the computed connectivity, neighbour relationships are compared with ground truth (GT) via the overlap ratio (Eq. 14):

$$O_i = \frac{\mid N_i^{\mathrm{comp}} \cap N_i^{\mathrm{GT}} \mid}{\mid N_i^{\mathrm{GT}} \mid} \quad \text{(Eq. 14)}$$

A ratio close to 1 confirms strong agreement with real structural relationships, ensuring accurate reconstruction of intersections, alignments and connectivity. This wall topology then supports downstream geometric reconstruction (Figure 29) and IFC model generation (Figure 29, right), although IFC export remains optional within the proposed pipeline.

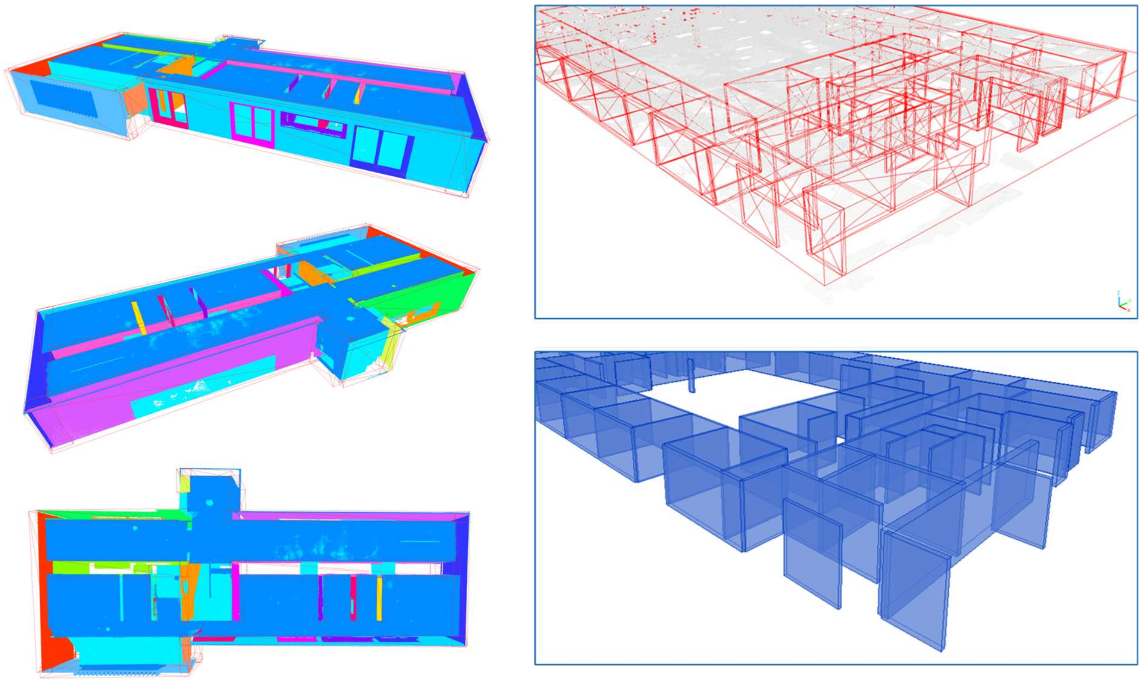


Fig. 29. Wall reconstruction. Left: semantic classification of point clouds and wall OBBs. Right: reconstructed wall surfaces aligned with the point cloud (top) and corresponding IFC wall elements generated from the reconstructed geometry (bottom).

4.2.5.4 Columns

Column instances, derived from the column clusters $\mathcal{L}_i$, are reconstructed through a multi-step procedure designed to overcome noise and recover rectangular geometry with high precision (Fig. 30). First, an oriented bounding box (OBB) is computed for each cluster. This first OBB includes peripheral noise points. The points of each cluster are then projected onto a xy plane (Fig. 31) at the average elevation $\bar{z}$, thereby flattening the cluster while preserving planimetric shape. The bounding box is aligned with the x and y axes by estimating a rotation matrix R and applying it before projection at height $\bar{z}$ (Eq. 15):

$$P_i^{\text{proj}} = (x_i', y_i', \bar{z}) \qquad \text{(Eq. 15)}$$

In this two-dimensional space, the convex support of the projected points is computed as a *Convex Hull* $\mathcal{H}$(P)(Eq. 16):

$$H(P) = \{\, p \mid p = \textstyle\sum_{k=1}^{N} \lambda_k p_k \,,\ \sum_{k=1}^{N} \lambda_k = 1,\ \lambda_k \geq 0 \,\} \qquad \text{(Eq. 16)}$$

To refine the initial box, each boundary segment of the previous OBB is partitioned into intervals (generally three intervals) and for each interval a Gaussian distribution is estimated from the point counts. The peak locations provide robust votes for the true supporting lines along x and y. Aggregating these votes yields the minimum oriented bounding box (mOBB) that best fits the data in the least noisy sense.

The mOBB is subsequently mapped back to the original three-dimensional position by applying the inverse rotation and translation, that is, a roto translation consistent with R and the cluster centroid. Column height is determined by the same procedure adopted for the wall class.

To ensure structural plausibility, clusters whose minimum side length is less than 0.15 m are not instantiated as columns. In summary, the Column geometry is obtained from the mOBB, providing width, length and height, with the column local origin placed at the center of the base face.

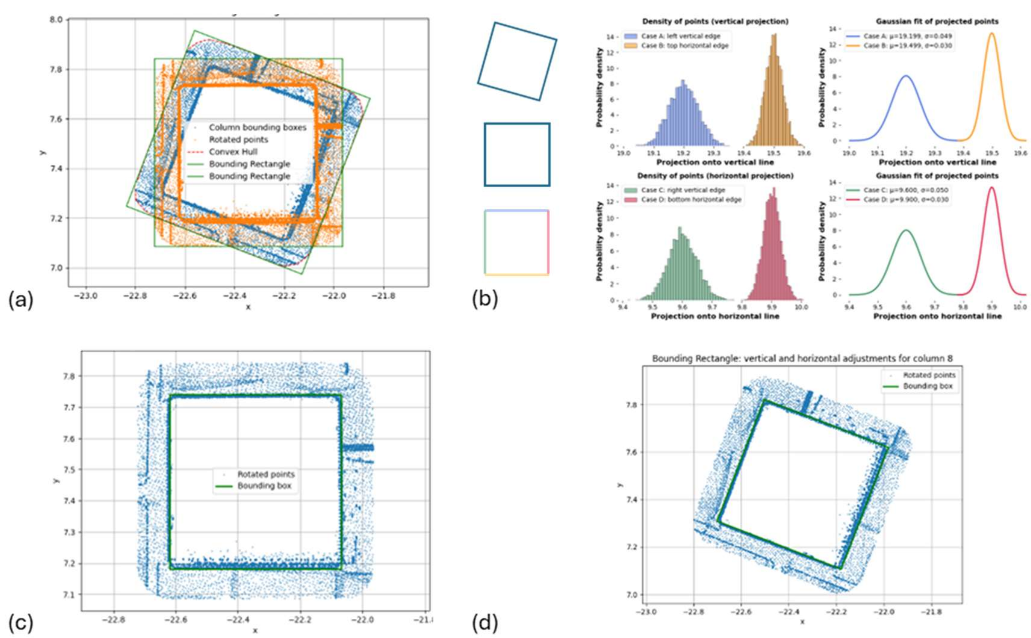


Fig. 30. Column cross-section reconstruction. (a) rotated point cloud projections and (b) corresponding density distributions with Gaussian fits for edge detection; (c) refined oriented bounding boxes obtained after vertical and horizontal alignment adjustments, and (d) final positioning.

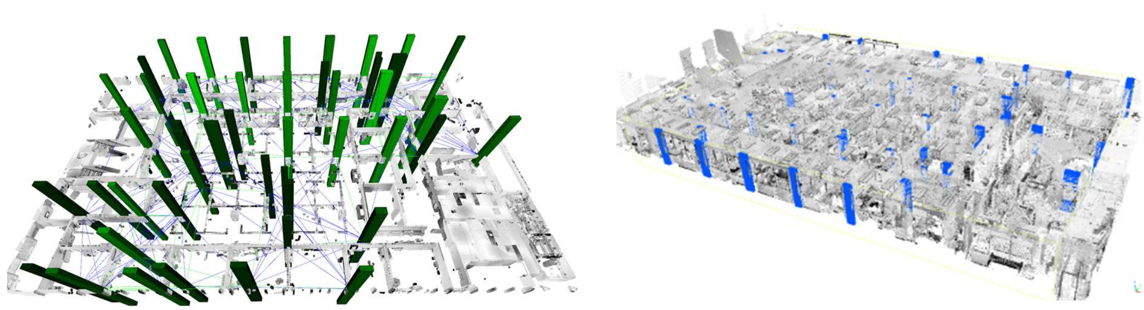


Fig. 31. Column 3D reconstruction (left) and spatial repositioning based on geometric refinement (right).

4.2.6 Openings: Doors and Windows

Door reconstruction is conditioned on the reconstructed walls that host the opening, and the same wall-anchored procedure is adopted for windows (Fig. 32).

Candidate openings are detected from wall nodes and wall point clouds by combining geometric cues with image-based evidence: in the data, an opening may appear either as a cluster of points outlining the frame and leaf or as a local void where the absence of points indicates a doorway or window.

Once a candidate is confirmed, its 3D geometry is instantiated by sampling the surface between the start point $P_s = (x_1, y_1, z_1)$and the end point $P_e = (x_2, y_2, z_2)$. The opening thickness is taken equal to the

supporting wall thickness, and the principal axis of the opening's oriented bounding box (OBB) is aligned with the wall axis to preserve orientation consistency. For evaluation, reconstructed openings are compared with reference annotations on geometric grounds.
The center of an opening is given by $c = \frac{1}{2}(P_s + P_e)$, and positional accuracy is measured via the Euclidean distance between reconstructed and reference centers, $d = \| c - c^* \|_2$. The opening area is computed from its OBB as (Eq. 17):

$$A = w \cdot h, \qquad w = | x_2 - x_1 |, \qquad h = | z_2 - z_1 |, \tag{Eq. 17}$$

with w the horizontal span and h the vertical span between the endpoints. Joint analysis of d and A validates both spatial placement and dimensional fidelity, ensuring that detected openings conform to real-world sizes and are correctly positioned within the reconstructed architectural model.

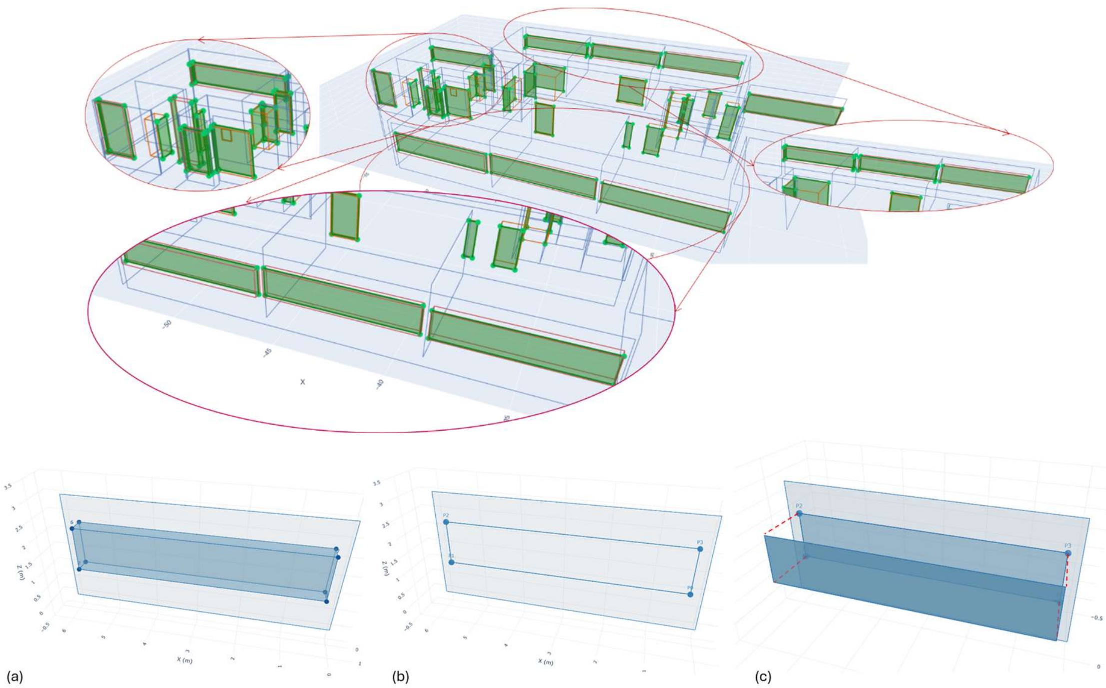


Fig. 32. Opening creations: (a) intersection points between wall surfaces and openings oriented bboxes (OBBs); (b) opening cut in the main wall surface; (c) opening surface creation.

The only semantic distinction between the two types is their elevation constraint: doors are required to have their base close to floor level, whereas windows must lie higher on the wall, with an average elevation typically greater than 0.45 m.
To refine the set of opening candidates (doors and windows) produced by the GroundingDINO network, the following geometric rules and post-processing steps are applied (Fig. 33).
GroundingDINO frequently produces several overlapping or adjacent bounding boxes within the same wall view, and because raster images derived from point clouds may be low-quality (e.g., due to sparse or noisy sampling), detection scores are commonly retained above 0.20 m; as a result, score values alone cannot reliably filter redundant detections, making additional geometric post-processing necessary. Distances along the horizontal and vertical image axes between adjacent detected openings are therefore evaluated: the separation along the x-axis (width) is denoted (δ_l), and along the y-axis (height) (δ_w).

If two bounding boxes are closer than 0.40 m in either direction, they are considered to represent the same physical opening and are merged into a single box whose extent is defined by the most external coordinates (*min x, max x, min y, max y*); otherwise, they are retained as separate detections. Subsequently, containment relationships are checked: if one bounding box lies entirely inside another, only the vertices of the larger box are kept, and the inner one is discarded. These post-processing rules are applied uniformly to both door and window detections.

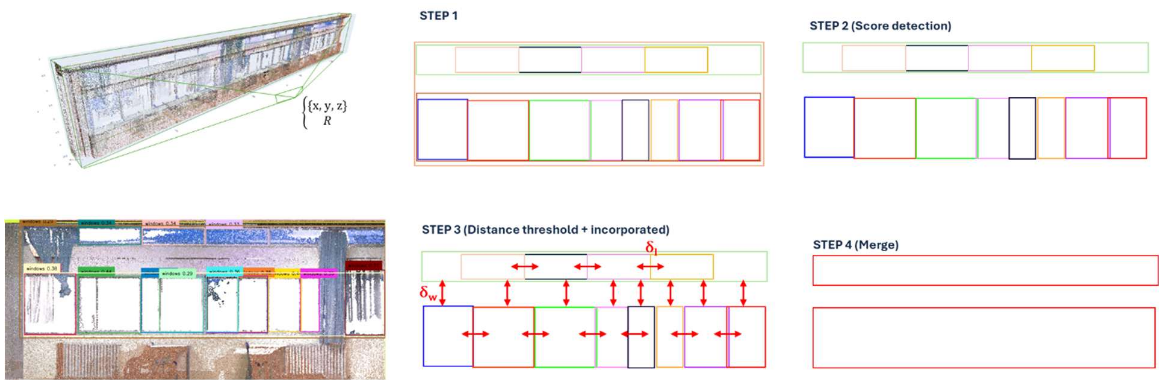


Fig. 33. Vision- and geometry-based opening reconstruction process. The workflow illustrates opening detection, distance-based validation and intersection handling, followed by merging into continuous opening element.

4.3 Device detection: Scan-to-BEM

To address the difficulty of detecting small devices in complex indoor environments, a lightweight interactive tool was developed to enhance the identification and mapping of devices and energy systems (Fig. 34).

The interface provides a 3D point cloud viewport with navigation and class selection (lighting, radiators, sensors). Device instances are selected by clicking near the object in the scene; a ray-casting step followed by a *k-d-tree* nearest-neighbor (NN) search snaps the pick to the closest in-class point within $r_{\text{snap}} = 0.10$ m. Each confirmed instance records the class name, element ID, and global coordinates (x, y, z)in meters. Using these coordinates, assignment to rooms or thermal zones is performed via a point-in-solid query; ties are resolved by the largest volumetric intersection, and a per-class minimum spacing of $d_{\text{min}} = 0.25$ m (for lights) is enforced to avoid duplicates.

To improve robustness, room and zone volumes are expanded by $\delta = 0.15$ m (approximately half a wall or ceiling thickness) before the point-in-solid test, and the element is then assigned to the corresponding zone. When clicking, in addition to the device class (lighting, radiators, sensors), a specific device typology can be selected based on efficiency. For lights and radiators, three efficiency levels (low, medium, high) are available; each level automatically retrieves the corresponding technical data (e.g., nominal power, efficiency parameters) stored in the Information Loading Dictionaries (ILDs) (Section 2.6).

Finally, annotations are exported to JSON with *class_name*, *element_id*, coordinates, *room_id* and confidence rate value. Although the workflow is manually driven, this spatially aware approach enables precise localization and classification of energy-critical components, such as recessed luminaires and embedded radiators, thereby enhancing the completeness and accuracy of energy performance modeling.

The final mesh model can optionally be exported in IFC format and is also serialized into JSON files using an explicit representation that includes vertex coordinates, rotation matrices, object centers, thicknesses, heights, semantic class labels, and unique object identifiers. A visual representation of the final models for walls, columns, and doors is shown in Fig. 35.

Fig. 34. Results of the developed tool for manual detection and classification of energy-related elements. (a) User interface with lighting elements highlighted, (b) detected lights, (c) detected radiators, (d) detected devices and sensors, and (e) top view of the scene after element detection.

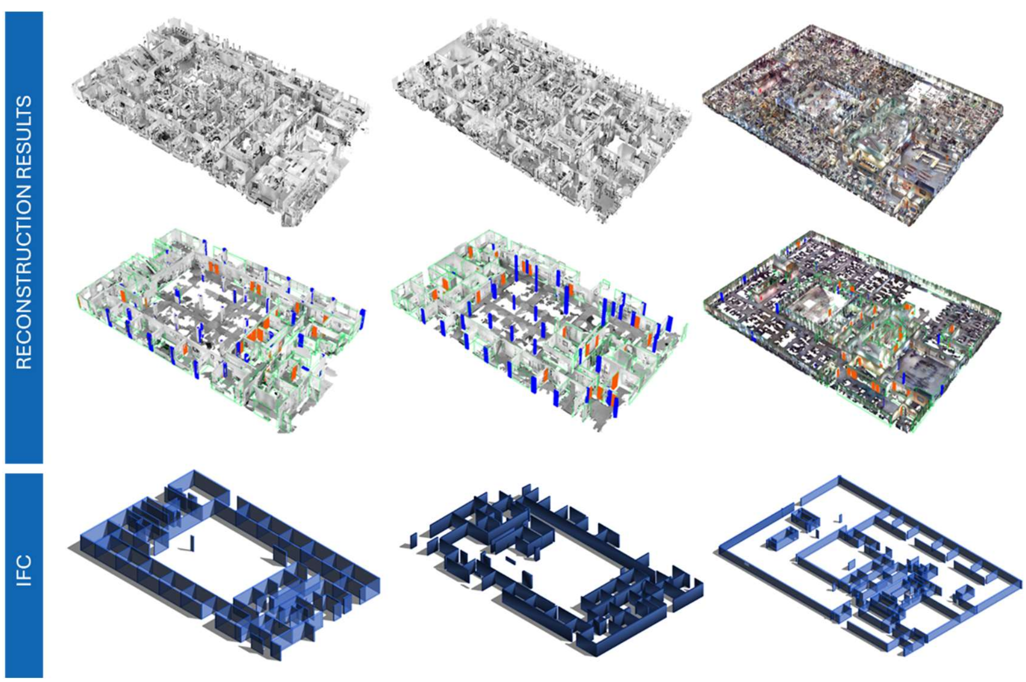


Fig. 35. End-to-end reconstruction results. Top: point cloud classification, structural element detection and reconstructed geometry. Bottom: resulting IFC-based wall models suitable for downstream processes.

4.4 Topologic model

The Topologic model (Section 2.7) is built on fundamental mathematical entities, from vertices up to *CellComplex* structures. Topology encodes the relationships and connectivity among these entities, forming the model's essential structural framework (Chen et al., 2022a; Arroyo Ohori et al., 20218). This structure can be naturally represented as a graph, where entities correspond to nodes and relationships to edges that capture adjacency, incidence, neighborhood and hierarchical containment (parent–child) relations, as detailed in the following sections.

Integrating the B-Rep representation, which is primarily geometric, within a topologic (relationship-based) model results in a structured, graph-oriented representation that combines geometric precision with topological intelligence, enabling advanced analysis and interoperability.

Furthermore, linking a B-Rep that defines thermal zones or room volumes and their constituent surfaces, semantically enriched with labels such as `thematic_surfaces` and including vertex coordinates, within a topologic framework creates a structured asset in which all entities are represented as a graph. This structure can then be enriched with external semantic information, such as material properties, transmittance values (*U-values*) and other physical attributes of the real asset, supporting deeper analysis and more effective data integration (Fig. 36).

However, the first step is to define the (i) volumetric model, derived from the 2D Topologic Map and the (ii) surface-based model, both of which must be linked and integrated into the main framework.

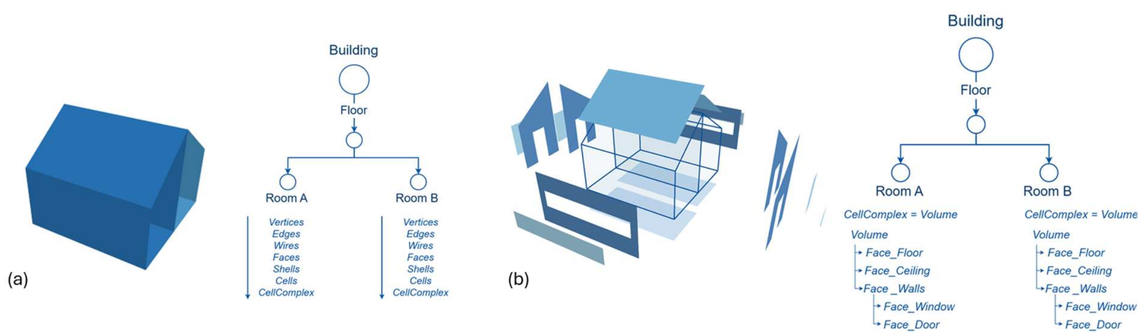


Fig. 36. Geometric solid to topology-aware semantic models. The figure illustrates the (a) solid room or thermal-zone geometry to a (b) hierarchical, topology-aware, surface-based semantic representation suitable for simulation and DT applications.

4.4.1 Topologic Map: the floorplan definition

The *Topologic Map*, first introduced in (Roman et al., 2024), represents a bidimensional, topological and semantically enriched reconstruction of a 2D floor plan. It comprises four elements:

1. *Vertices:* produced by the intersection-looping algorithm (Sec. 3.5.1.1); they correspond to wall–wall intersections in the 2D xy plane.
2. *Segments:* edges connecting pairs of vertices; they are validated by checking geospatial overlaps with the input oriented bounding boxes (OBB) from the classified point cloud (Fig. 37 defined as supported/not supported by OBB) of walls detected in the point cloud (Sec. 3.5.1.2).
3. *Half-edges:* a directed version of a segment used in the Doubly Connected Edge List (DCEL) structure.
4. *Twins:* the opposite-direction half-edge of the same segment. The half-edge-twin pair (i.e., a double-connected representation) enables DCEL traversal of edge paths to derive face polygons.

All elements are assigned a unique identifier and the same applies to every half-edge and its twin. This guarantees referential integrity and is required to maintain a manifold mesh with well-defined polygonal

faces. Furthermore, at each intersection, segments are split; consequently, within a single OBB of the point cloud there are often multiple half-edge and twin pairs rather than a single pair.
The implementation of the DCEL algorithm enables a semantic-aggregation step.
Specifically, by checking whether a polygonal face is associated with the twin of each segment, it becomes possible to classify segments, and therefore walls, as internal or external. Segments whose twins are adjacent to a polygonal face are considered internal, while those without an adjacent face are classified as external. The classification is propagated to the 3D representation, allowing rule-based assignments: (i) exterior materials are applied exclusively to external walls; (ii) interior finishes are applied only to internal walls; (iii) openings on external walls are instantiated as windows, consistent with common BIM conventions; and (iv) openings on internal partitions are instantiated as doors.

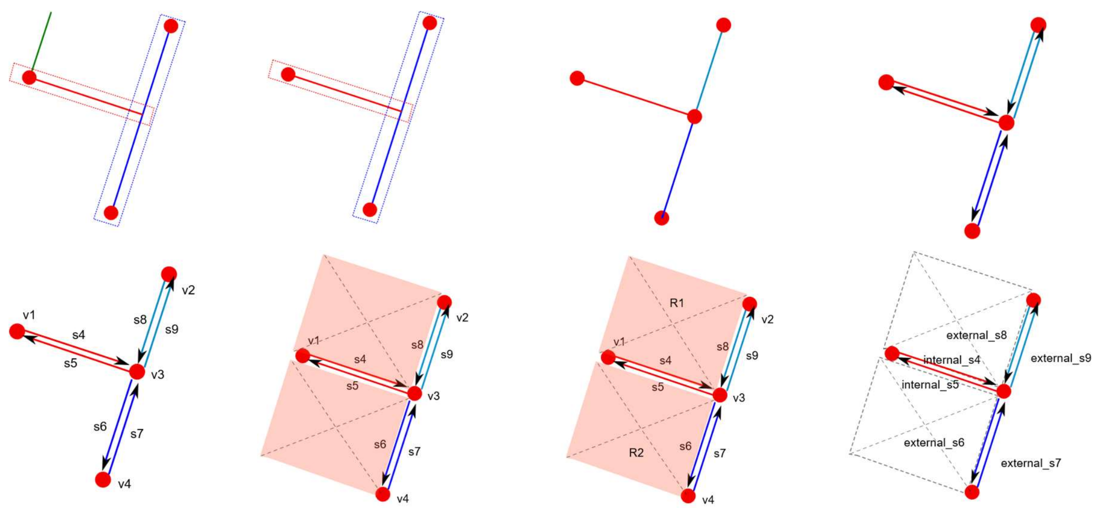


Fig. 37. DCEL implementation: a. Vertices in red, segment supported by OBB in red and blue, while in green not OBB supported segment. b. Only the OBB supported walls. c. Split segments in intersection points. d. Double connected segments.

Mathematically, the 2D *Topologic Map* is defined as follows.
Let $\mathcal{P} \subset \mathbb{R}^3$be the input point cloud, from which a set of wall-oriented bounding boxes (OBBs) defined as $\mathcal{W} = \{\text{OBB}_1, \dots, \text{OBB}_m\}$ is extracted. The wall–wall intersections in the xy-plane define the set of intersection vertices $\mathcal{V} = \{v_i \in \mathbb{R}^2\}$.
A segment is a 1–simplex $s_{ij} = (v_i, v_j) \in \mathcal{S}$, and it is said to be supported by an OBB if:

$$\exists\ \text{OBB}_k \in \mathcal{W}\text{: overlap } (s_{ij}, \text{OBB}_k) = 1. \qquad \text{(Eq. 18)}$$

Segments are split at intermediate vertices. Each segment s_{ij} is represented by two directed half-edges $(h_{ij}: v_i \to v_j)$ and $(h_{ji}: v_j \to v_i)$ with the twin relation twin $(h_{ij}) = h_{ji}$. Together with the operators next (h), prev (h), and face (h), these half-edges form a DCEL. Faces are cycles of half-edges, $F = (h_1, \dots, h_k) \in \mathcal{F}$. A segment $s_{ij}$is classified as internal if its twin half-edge belongs to a face,

$$\text{isInternal } (s_{ij}) = 1 \iff \text{face } (h_{ji}) \neq \emptyset, \qquad \text{(Eq. 19)}$$

and as external otherwise. Let $f: \mathcal{S} \to \mathcal{B}$ map segments to 3D wall elements. The material and opening types are then assigned by

$$material = \begin{cases} interior, & \text{isInterna l}(s) = 1, \\ exterior, & \text{isInterna l}(s) = 0, \end{cases} \quad \text{(Eq. 20a)}$$

$$opening\big(f(s)\big) = \begin{cases} door, & \text{isInterna l}(s) = 1, \\ window, & \text{isInterna l}(s) = 0. \end{cases} \quad \text{(Eq. 20b)}$$

4.4.2 Iterative intersection algorithm

To reach the *Topologic Map* definition (Section 4.1.12), the first step is to detect and compute all wall intersections.

The algorithm ingests a *Walls* ($\mathcal{W}$) class containing walls, their centerlines and openings (doors, windows), projects centerlines to 2D, removes sub-threshold segments and consolidates near-duplicates via angular and perpendicular-distance tolerances, retaining the longer segment (length threshold $l_{th} = 0.35\ m$).

It then iteratively extends lines with increasing scale factors ($step = 0.05\ m$) and computes all pairwise intersections to assemble a deduplicated set of junctions (Fig. 38). The algorithm includes an early-termination guard on each line extension: if any extended segment finds more than two intersections, it is deemed to be extending beyond its plausible wall extent and is stopped.

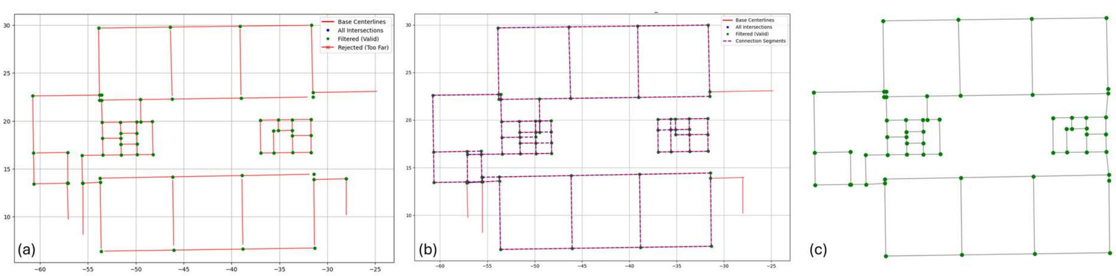


Fig. 38. Intersection computation and edge refinement process: (a) iterative computation of all intersections; (b) retention of overlapping segments only; and (c) cleaned and refined floorplan segments.

This is then double-checked by testing the segment against the wall oriented bounding boxes (OBBs) from the graph in Section 4.1.4: the segment must lie entirely within the corresponding OBB boundaries; otherwise, it is clipped or discarded. Together, these checks prevent overextension artifacts and keep intersections confined to the true wall extents.

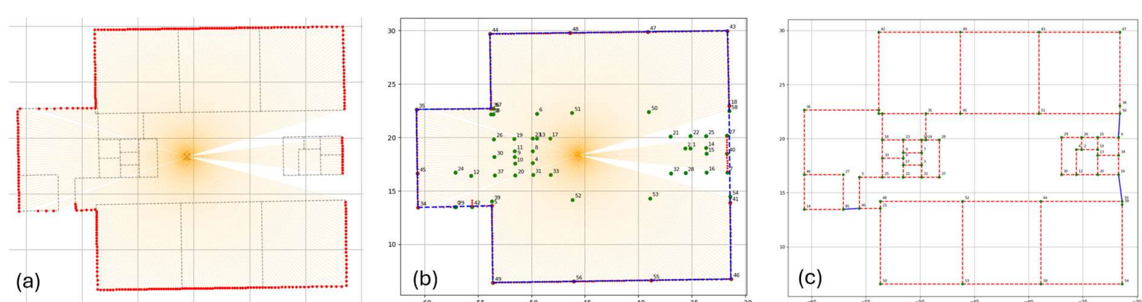


Fig. 39. External segment identification: (a) ray tracing for external edge detection; (b) computation of external segments; and (c) final cleaned edges and closure segments (highlighted in blue).

Finally, intersections are filtered by proximity to base centerlines and by local support (neighbor density), and each accepted point is assigned to its nearest extended line. The output is a robust 2D NetworkX graph of wall centerlines with validated intersection points (Fig. 39).

4.4.3 Double connected Edge List (DCEL)

To build a room DCEL from wall centerlines, the pipeline first polygonizes the centerline graph using `polygonize_full`, separating valid cycles (candidate rooms) from open chains (cuts and dangles). Sub-threshold rooms (area $A_{\min} < \ell_1 \cdot \ell_2 = 0.90\,\text{m} \times 0.90\,\text{m}$, i.e., the minimum footprint to accommodate a standard door swinging either direction) are merged into larger adjacent rooms. Exterior rings are then extracted and a DCEL is generated in which each boundary edge is represented by paired `half-edges` with mutual `twins`, consistent `next/prev` links, and correct incident faces (Fig. 40).

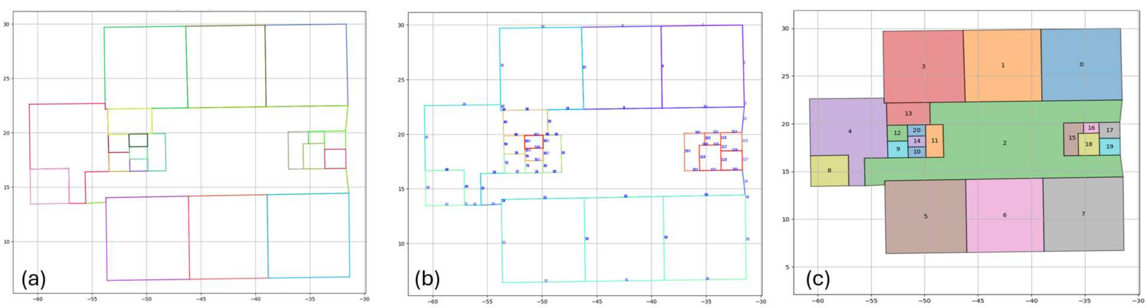


Fig. 40. Floorplan generation workflow: (a) extracted individual edges; (b) labeled and classified edges; and (c) resulting room polygon used for floorplan definition.

The DCEL construction yields polygonal faces $\mathcal{F} = \{f_i\}_{i=1}^{n}$, where each f_i represents one room cell.


```
{
  "vertices":[
    {"id":"v0","x":0.18,"y":0.22,"incidentHalfEdge":"e0"},
    {"id":"v1","x":0.84,"y":0.26,"incidentHalfEdge":"e2"},
    {"id":"v2","x":0.9,"y":0.86,"incidentHalfEdge":"e4"},
    {"id":"v3","x":0.32,"y":0.96,"incidentHalfEdge":"e6"},
    {"id":"v4","x":0.1,"y":0.56,"incidentHalfEdge":"e8"}],

  "halfEdges":[
    {"id":"e0","origin":"v0","twin":"e1","next":"e2","prev":"e8","face":"f1","sem":{"bound
    ary":"external"}},
    {"id":"e1","origin":"v1","twin":"e0","next":"e9","prev":"e3","face":"f0"},
    {"id":"e2","origin":"v1","twin":"e3","next":"e4","prev":"e0","face":"f1","sem":{"bound
    ary":"external","highlight":true}},
    {"id":"e3","origin":"v2","twin":"e2","next":"e1","prev":"e5","face":"f0"},
    {"id":"e4","origin":"v2","twin":"e5","next":"e6","prev":"e2","face":"f1","sem":{"bound
    ary":"external","highlight":true}},
    {"id":"e5","origin":"v3","twin":"e4","next":"e3","prev":"e7","face":"f0"},
    {"id":"e6","origin":"v3","twin":"e7","next":"e8","prev":"e4","face":"f1","sem":{"bound
    ary":"external"}},
    {"id":"e7","origin":"v4","twin":"e6","next":"e5","prev":"e9","face":"f0"},
    {"id":"e8","origin":"v4","twin":"e9","next":"e0","prev":"e6","face":"f1","sem":{"bound
    ary":"external"}},
    {"id":"e9","origin":"v0","twin":"e8","next":"e7","prev":"e1","face":"f0"}],

  "faces":[
    {"id":"f1","outerComponent":"e0","innerComponents":[],"properties ":{"type":"room"}},
    {"id":"f0","outerComponent":null,"innerComponents":["e1"]}]}
```

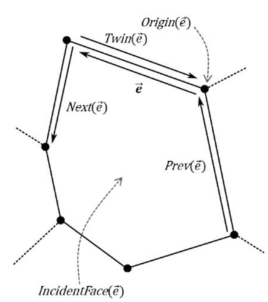



Table 12. DCEL definition and implementation, illustrating the construction from vertices to half-edges and twin relations, and finally to polygonal faces.

Boundary segments from the floor's *Topologic Map* are assembled into closed polygonal cycles, each cycle defining a room. Each room cell carries a unique ID and its boundary edges are annotated as internal or external, as shown in Section 4.4.3, while storing half-edge identifiers together with their twin references. The resulting model is:

(i) *watertight* because every boundary is closed,

(ii) it is *semantically enriched* because the room ID and each wall type with its ID is stored with geometry,

(iii) *manifold:* for every internal wall shared by two rooms, two coincident edges exist, one per room, with identical geometry and opposite orientation, meaning each edge's local normal points inward to its own room while its twin points into the adjacent room.

Openings on internal walls are represented in the same two-sided manner, with matching geometry and opposite orientation for the two adjoining rooms. This two-sided, orientation-aware encoding guarantees manifoldness and supports robust downstream operations such as meshing, extrusion, and volumetric reconstruction.

4.4.4 DCEL: internal and external walls

The DCEL structure provides a robust topological method also for classifying walls as internal or external, overcoming the geometric uncertainty of ray-tracing approaches. In this representation, every wall segment is modeled by a pair of half-edges (an edge and its twin), each optionally associated with an incident face. A wall is then defined as (Fig. 41):

- *internal* when both half-edges bound distinct bounded faces, indicating adjacency between two enclosed rooms;
- *external* when only one of the two half-edges is incident to a bounded face, while the opposite side is unbounded or assigned to the exterior face f_0.

Cases where neither side is bounded or both reference the same face are treated as ambiguous or degenerate geometries. This topological criterion directly leverages DCEL's adjacency relations, ensuring consistent wall classification independent of geometric sampling errors or incomplete ray-casting coverage.

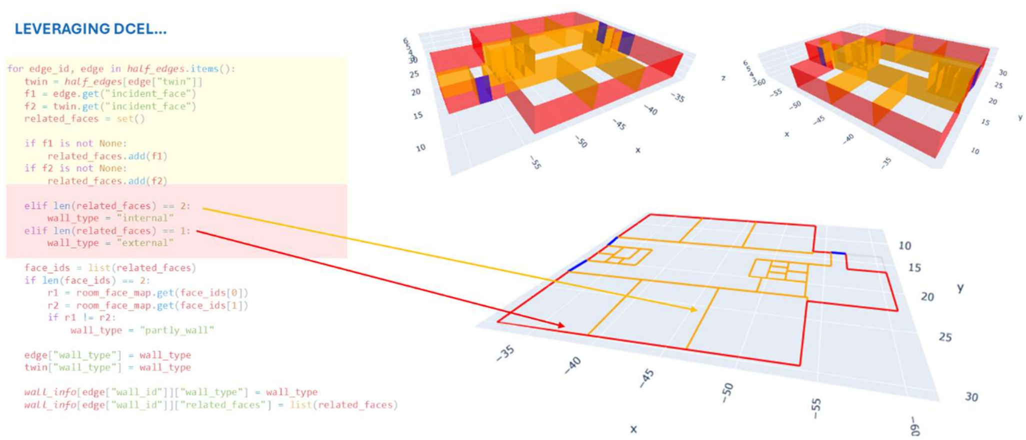


Fig. 41. Implementation of the DCEL-based classification algorithm, illustrating the identification of internal and external walls through half-edge and face adjacency analysis.

4.4.5 DCEL: floorplan checks and validation

To verify the final DCEL (Fig. 42), a validity pass checks half-edge reciprocity, reverse-geometry equality, and distinct incident faces. Occasionally, some edges are assigned to one polygon but not its opposite. This can occur when very small or nearly coincident edges are omitted during processing, typically due to (i) numerical tolerances, (ii) rounding errors, or (iii) insufficient edge separation, which hinder their correct registration in the DCEL structure and prevent the formation of valid polygon loops, leaving some edges orphaned.

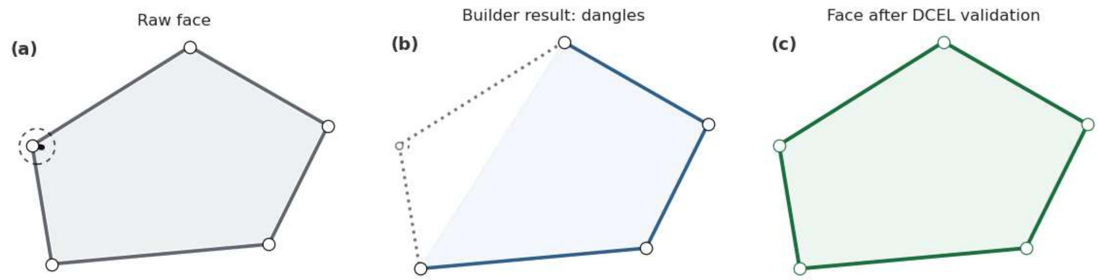


Fig. 42. Face validation using DCEL. (a) Raw, incomplete face, (b) intermediate dangling edges, and (c) corrected check and reconstructed polygonal face.

In the final cleanup stage, remaining dangles, such as open or disconnected line ends, are addressed: endpoints are snapped to nearby junctions within a snap tolerance $\varepsilon = 0.05\, m$, colinear segments are bridged under angle $\alpha = 5^{\circ}$ and offset $\delta = 0.05\, m$ tolerances and short gaps are closed up to a gap threshold $\tau = 0.10\, m$, with all repairs constrained by wall OBB bounds (Sec. 4.1.4) to prevent overreach.
The result is a clean, dangle-free 2D DCEL of room faces, with repaired edges labeled in semantic ({"repair": "gap_close"}) for traceability, that are later integrated in the dataframe with the correct semantic.

4.4.6 Volumetric topologic model

The volumetric model is generated by extruding the Topologic Map, which can be defined as its three-dimensional counterpart. To produce a volumetric model that is (i) watertight and (ii) topologically consistent, as in the B-Rep surface-based Topologic model discussed in the next section, the algorithm leverages TopologicPy[4]. TopologicPy is a spatial modeling and analysis library built on a C++ topology (Open CASCADE[5]) core with Python bindings.
It enables hierarchical, information-rich 3D representations that integrate geometry, topology, metadata and AI, enhancing BIM with building-intelligence constructs. The library supports scripting (CLI/Python), visual data-flow plugins for major BIM tools and interoperability through IFC, OBJ, BREP, HBJSON and CSV formats.
Following the hierarchical topological framework defined in TopologicPy, the model organizes geometric entities according to increasing dimensional complexity as:

$$Vertex \rightarrow Edge \rightarrow Wire \rightarrow Face \rightarrow Shell \rightarrow Cell \rightarrow CellComplex$$

thereby maintaining explicit topological relationships and enabling consistent adjacency, connectivity and containment reasoning across all levels of the representation. The component's *z_height* is inherited from the wall elements corresponding to each polygon defined by the DCEL algorithm. When walls enclosing a room have different *z_values*, for example when two adjacent rooms have ceilings of different heights, the algorithm:

- performs a nearest-neighborhood (NN) search to read the heights of the walls enclosing each room;
- aggregates the retrieved heights;
- selects the most frequent (modal) z value within the room;
- extrudes the room volume up to that computed z.

[4] https://topologic.app/topologicpy_doc/
[5] https://dev.opencascade.org/

This procedure resolves inconsistencies arising from adjacent rooms (or cells) with different ceiling heights by exploiting the manifold properties of the DCEL-derived room polygons. The algorithm also records geometric features derived from the volumetric outputs, specifically the computed volume of each room and the set of vertices defining its extreme boundaries, which are useful for subsequent geometric assessment and accuracy analysis of the reconstructed model (Fig. 43).

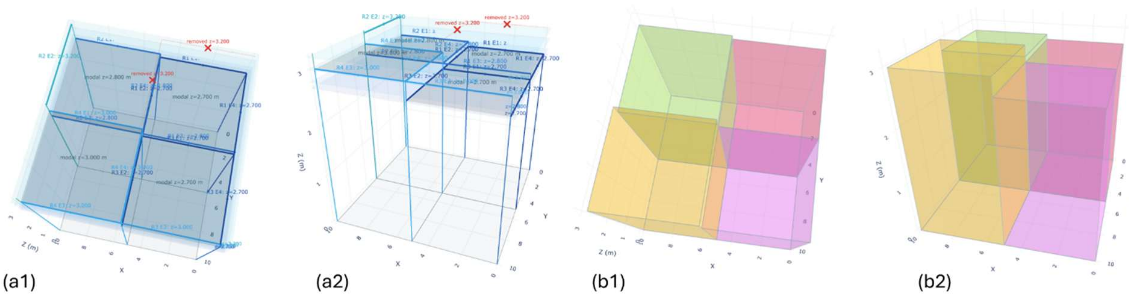


Fig. 43. DCEL-based room volume extrusion and height resolution. When enclosing walls exhibit different z-values, room heights are determined via nearest-neighbor (NN) wall sampling and modal height selection, ensuring topological consistency and correct volumetric extrusion across adjacent rooms.

4.4.7 Surface-based topologic model

The surface-based B-Rep model describes the topological structure (Fig. 44) of the reconstructed geometry by defining only the manifold surfaces of the model. It does not decompose the previously derived solid volumes; instead, starting from the centerlines of walls and openings such as doors and windows, it reconstructs the building envelope as if the volumes were exploded into their faces while preserving all semantic information.

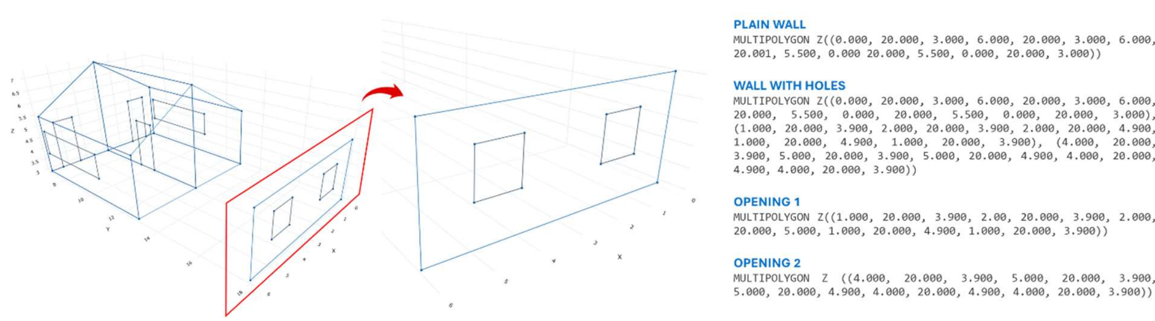


Fig. 44. Wall reconstruction with opening extraction. The figure illustrates the WKT-based definition of the resulting topological model.

Openings are identified through two-dimensional intersections in the xy plane with wall centerlines and assigned accurate elevation values obtained from the RDF graph. Each wall is then transformed into a local coordinate system with origin at (0,0) in the working xz plane using its rotation matrix $\mathcal{R}$. In this frame, the opening rings are computed to define the exact geometric footprint of windows and doors, which are subtracted from the wall surfaces by Boolean operations before being transformed back to global coordinates. This ensures consistency of the process and robustness of the results.

Floors and ceilings are reconstructed from their base and top edge loops using nine-decimal precision to ensure watertight and manifold continuity. All geometry is exported as three-dimensional WKT with explicit definitions, using MULTIPOLYGON Z and LINESTRING Z geometries (Fig. 45).

Walls are represented as polygons with interior rings corresponding to openings, centerlines and opening perimeters are stored as line strings, and floors and ceilings are defined as polygons reconstructed from consolidated edge loops.

Fig. 45. Custom DataFrame schema for the topological representation of reconstructed building elements, storing hierarchical relations, thematic surface labels, WKT geometry and connectivity information.

Each WKT feature carries a stable identifier and links back to the volumetric model: every wall is a child of its room (cell), each room is nested within its story and then the building, and explicit adjacencies are maintained to the room's walls, floor and ceiling; any opening is a child of its host wall. This establishes a clear hierarchy with consistent parent–child and peer relations, while room-to-room connectivity is reinforced via a DCEL-style structure that records twin and half-edges, ensuring robust adjacency, incidence and containment across the model.

4.5 Topologic graph

The graph is constructed directly from the CSV output by treating each row as a node representing a unique building element, identified by *element_id*. Hierarchical relations are derived from *parent_id*, which encodes containment and enables reconstruction of the building's spatial structure (Building → Storey → Room → Surface → Edge). The *thematic_surface* field specifies the semantic type of each element (e.g., wall, floor, ceiling, opening), allowing class assignment and supporting downstream material and construction mapping (Fig. 46).

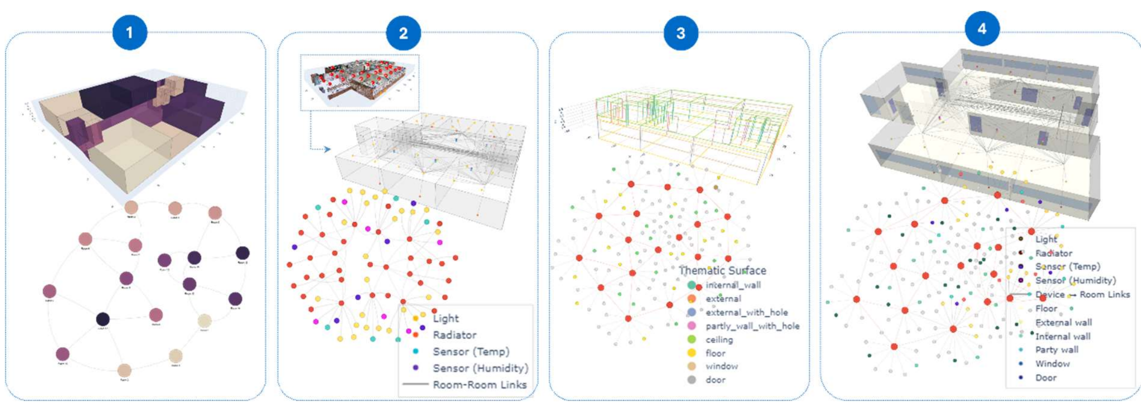


Fig. 46. Steps of graph enrichment: (1) Topologic solid model with rooms; (2) solid model enriched with energy-related elements; (3) B-Rep model with thematic surfaces; and (4) complete graph integrating thematic surfaces and energy elements.

Geometric information is incorporated using *geometry_wkt*, which provides the element's 2D or 3D geometry in Well-Known Text format; wall polygons flagged by *has_holes* are treated as perforated surfaces (e.g., walls with openings), with the WKT geometry encoding both external and internal rings. Additional associative relations, such as adjacencies or references to other elements, are obtained from the *related_to* field and encoded as typed edges. Once parsed, all nodes and edges are assembled into a typed-property multigraph that preserves hierarchy, semantics, geometry and topological consistency, forming the backbone of the multi-level DT.
Steps to derive the graph structure are:

- The CSV output is first parsed and each row is interpreted as a building element, identified by *element_id*, with *parent_id*, *thematic_surface* and *geometry_wkt* providing hierarchy, semantics and geometry.
- Room cells are generated by selecting rows with thematic_surface = "*room*", collecting all child surfaces (*parent_id* = *room_id*), converting their *geometry_wkt* polygons to Topologic faces, and assembling them into shells and then volumetric cells (CellComplex). In particular, vertices are extracted from geometry_wkt and topology and geometry are assembled:

Vertex → Edge → Wire → Face → Shell → Cell → CellComplex

- For each room cell, a centroid vertex is computed and annotated via a Topologic dictionary (key ="room" and an index), providing a compact representation of room locations for graph construction.
- Thematic surfaces (e.g., external walls, party walls, floors, ceilings, windows, doors) are similarly converted from *geometry_wkt* to faces and enriched with "*id*" and "*type"* attributes, preserving semantic labels from thematic_surface.
- Building devices (lights, radiators, and environmental sensors) are instantiated from predefined 3D coordinates, each encoded as a vertex with attached metadata ("*id*", "*type*" and "*subtype*" for temperature/humidity sensors) using Topologic dictionaries.
- Each device vertex is connected to its closest room centroid by an edge, establishing device–room relationships and enabling room-level aggregation of sensor readings and actuation.
- Room–room connectivity is modelled by linking each room to its k-nearest neighboring rooms (k =12) via edges, resulting in an adjacency network that supports path, connectivity and zoning analyses.
- All room vertices, device vertices and edges are then assembled into a typed property multigraph (Topologic + NetworkX), in which nodes carry semantic and geometric attributes and edges encode containment, proximity and functional relations.

Finally, Fig. 46 shows 4 main steps of graph enrichment from (1) room nodes to (4) complete topologic DT.

4.6 Multi-level geometric Digital Twin

In this work, the multi-level DT is conceived as a geometric model with two structured levels and an accompanying graph-based data structure that jointly encode geometry, topology, and semantics in a unified representation. This framework enables a broad range of downstream tasks, including monitoring, analysis and simulation, by providing a coherent multi scale digital representation of the built environment. From a geometric point of view, the DT is instantiated both as a volumetric B-Rep model (Fig. 47a) and as a surface-based B-Rep model (Fig. 47b), to create a multi-level DT (Fig. 47c). Figure 48 show the same representations for a more complex building.

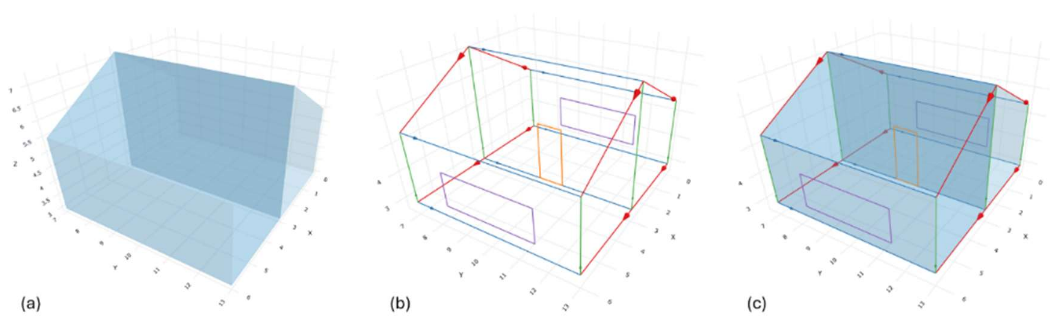


Fig. 47. Multi-level DT representations: (a) volume-based model; (b) thematic surface–based model; and (c) combined multi-level representation integrating volumes and thematic surfaces.

The final 3D structure is a multi-level topological representation that couples volumetric cells with their bounding surfaces, so that both space and its enclosing geometry are explicitly encoded. At the volume level, rooms are modeled as closed 3D cells obtained by extruding their 2D floor polygons between floor and ceiling; these cells tessellate the interior and are linked by shared faces that encode room–room and room–exterior adjacencies, enabling connectivity, neighborhood and circulation queries.

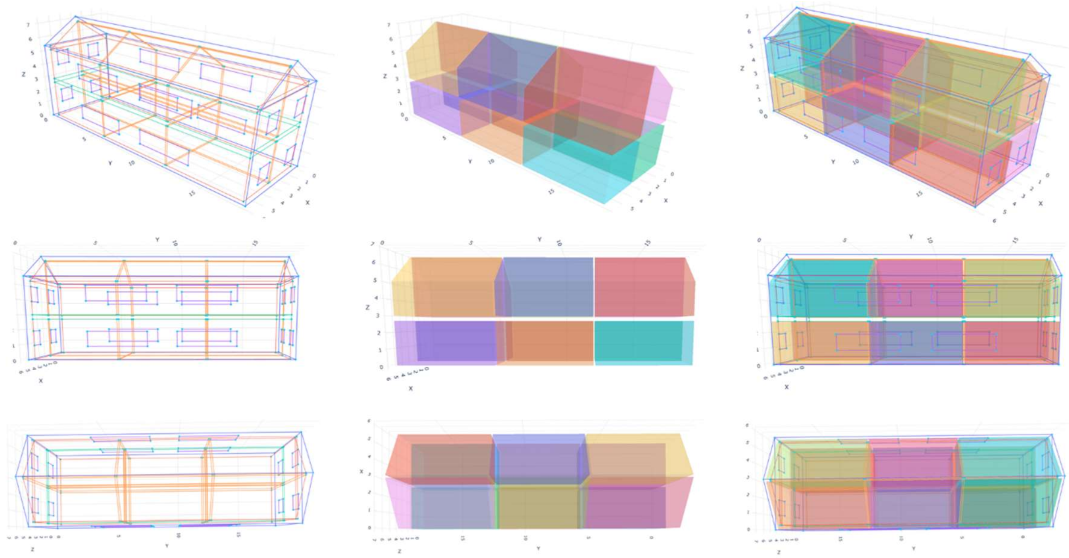


Fig. 48. Multi-level DT representations extended to a building configuration with an inclined roof.

At the surface level, each wall, floor and ceiling is represented as an individual polygonal face attached to its parent room. Faces are consistently oriented, share edges and vertices where appropriate and are stitched into a manifold assembly without gaps or overlaps. This surface layer is the natural support for attaching semantics (e.g. materials, functions, doors and windows) and for running downstream analysis, rendering and simulation, while remaining fully consistent with the volumetric topology above.
The multi-level DT (Fig. 49) is thus a structured model that is geometrically derived from Scan-to-BIM conventions for the z component, and from the Topologic model for semantics, topology, adjacencies and consistency.

4.7 Leveraging graph structure

The graph structure (Section 2.8) associated with the geometric model mirrors the hierarchical organization of the multi-level DT. The building entity contains its constituent storeys, each of which aggregates a set of rooms. Every room is defined by its bounding surfaces, such as walls, floor and ceiling, and each wall in turn hosts its openings, such as windows and doors.
In addition, this structure makes it possible to:

- attach further information through external Information Loading Dictionaries or specialized libraries, for example (i) material assignment, (ii) definition of wall and external shell layer compositions, (iii) assignment of energy related parameters, and (iv) specification of transmittance values and national regulation setpoints (Section 4.6.1);
- support interoperability by transforming the Scan-to-EDT outputs into IFC, gbXML and epJSON through in house developed Parser–Transformer–Writer (PTW) algorithms (Section 4.7 and subsections);

- further leverage PTW to run energy simulations with dynamic material assignment and to test and compare multiple design or operational scenarios (Section 4.8).

4.7.1 Information Loading Dictionaries (ILD) and external Libraries

In this framework, Information Loading Dictionaries (ILDs) function as JSON-based semantic containers or libraries linking each geometric or topological element to its operational, physical and regulatory metadata for the existing building (Section 2.9). In the Scan-to-EDTs framework, ILDs encode several types of data:

- the associations between room types and national rule sets, such as prescribed winter and summer setpoints, enabling automatic assignment of thermal parameters during scenario generation and checking process.
- They also manage the hierarchical structure of the material and construction libraries: individual materials, including their thermal and physical properties, are stored in a *material_library*, while *layer_construction* objects define assemblies composed of ordered material *layers* with specified thicknesses. Building elements (walls, floors, ceilings, and openings) reference these construction assemblies through their ILDs, allowing U-values to be computed directly from the material properties and layer definitions.

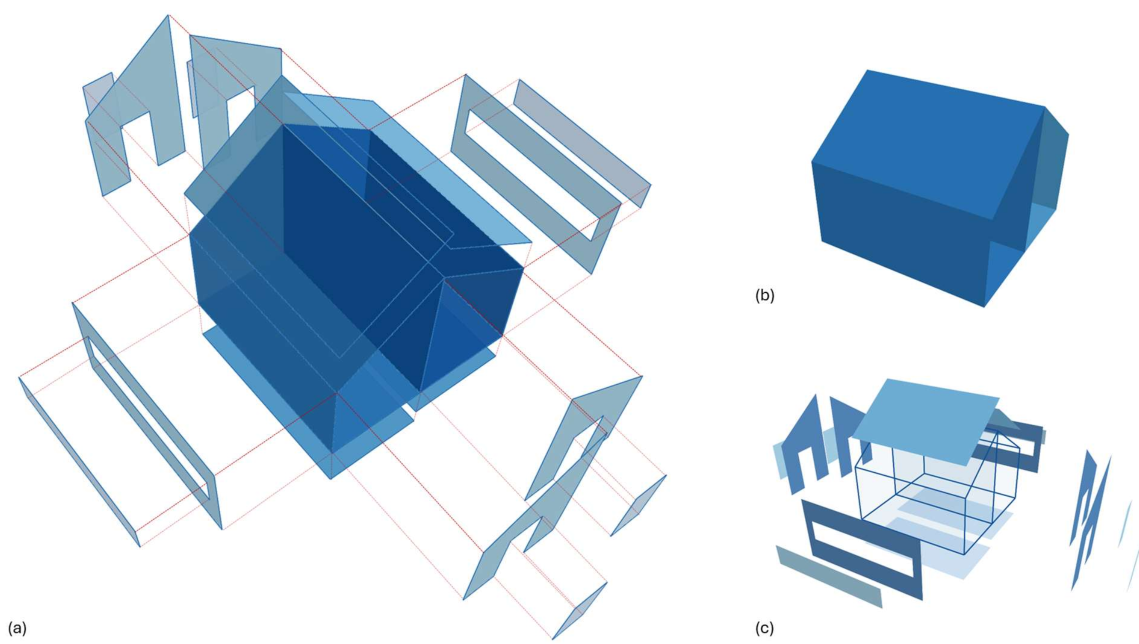


Fig. 49. Multi-level DT: (a) Decomposition into thematic surfaces; (b) solid volume representation; and (c) topology-aware semantic model with explicit surface and structural components.

- For openings, a specific ILD defines three levels of efficiency for windows (low, medium and high), each associated with a different U-value, while a single average U-value is assigned to all indoor doors.
- For devices a specific ILD defines the technical characteristics of radiators and lights (Section 4.2), including their typology and efficiency-dependent parameters.

By embedding these links as JSON dictionaries attached to nodes and surfaces, ILDs provide a consistent mechanism to propagate semantics, material compositions, setpoints and energy-related attributes across the DT, ensuring traceability, automatic schema mapping and interoperability. Thanks to the underlying triple-based structure, additional or more detailed properties can be easily introduced or updated without changing the core model.

4.7.2 Material assignment

Material assignment for opaque elements is driven by wall thickness.

First, all wall instances are isolated from the *thematic_surface* column and their original oriented bounding boxes (OBB) store in the DF graph (Section From point cloud to RDF graph (1)), including thickness, are imported. For each reconstructed wall surface, a centroid is computed and matched to the closest original wall bounding box. Then, if an overlap is found, the corresponding thickness is attached as metadata to that thematic surface. The resulting thickness values are then rounded to the nearest 0.05 m to enable robust matching with predefined construction packages in the material library.

The algorithm then allows the user to choose between brick-based or concrete-based walls, so that, once the thickness is matched, the corresponding *U*-value from the selected category is recalled from the *layer_construction* and assigned automatically to walls. Manual refinement is also supported, enabling overrides such as changing the material for individual elements when needed.

In the actual version of the algorithm (Fig. 50), floors, ceilings and roofs require a user-defined input value, whereas for openings it is sufficient to specify an efficiency level (low, medium, or high).

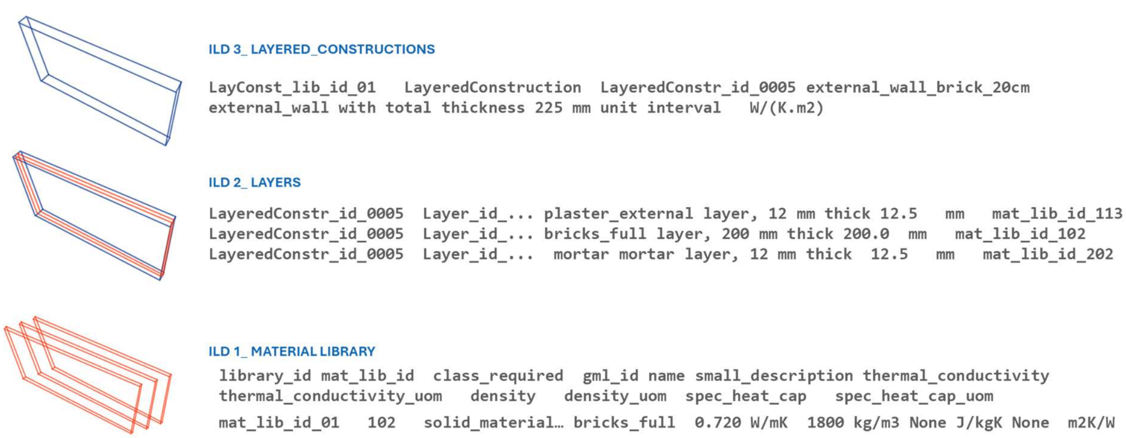


Fig. 50. Construction modeling workflow illustrating the material library, layer definitions, and resulting layered wall construction used for energy simulation and DT applications.

4.7.3 Parser-Transformer-Writers

The proposed graph model provides a multi-layered organization comprising: (i) a hierarchical layer for containment and inheritance; (ii) a topological layer describing areas, volumes, and adjacencies; (iii) a semantic layer encoding construction materials, assemblies, devices, and properties; and (iv) a compliance layer parameterized by national energy codes and standards. The model core is implemented as a SPARQL-based graph, enabling standardized querying, inference, and interoperability.

After assigning materials, device parameters, setpoints, and U-values, the enriched data are exported to an extended CSV format with additional attribute fields (highlighted in red, Fig. 51). This tabular representation is directly linked to the underlying graph through a SPARQL–SQL bridge, allowing the graph to be mirrored and manipulated as a *DataFrame* while preserving semantic consistency.

The resulting graph is high performance, standards-aligned with AEC workflows and designed for scalability and extensibility (Fig. 52). The multi-level DT is implemented as a typed-property multigraph (`NetworkX MultiDiGraph`) persisted in PostgreSQL and mirrored in RDF to support SPARQL querying.

(a)

(b)

Fig. 51. DataFrame-based enrichment workflow. The initial topological representation (a) is augmented with construction and thermal attributes (b).

Volumetric and surface models are unified in the multi-level DT through this typed-property multigraph. The graph encodes a strict hierarchy (*Building, Storey, RoomShell, Face, Edge, Vertex*) with integrated surface- and volume-level topology. Containment and incidence are modeled as directed acyclic edges, while adjacency is represented by reciprocal edges. Nodes and edges carry stable identifiers and descriptive attributes (e.g., materials and U-values). Face–RoomShell relationships mirror DCEL bidirectionality, enabling reliable adjacency and internal/external wall detection (Fig. 40). The graph is additionally mirrored in RDF using a lightweight schema to support SPARQL querying, preserving semantic and topological consistency.

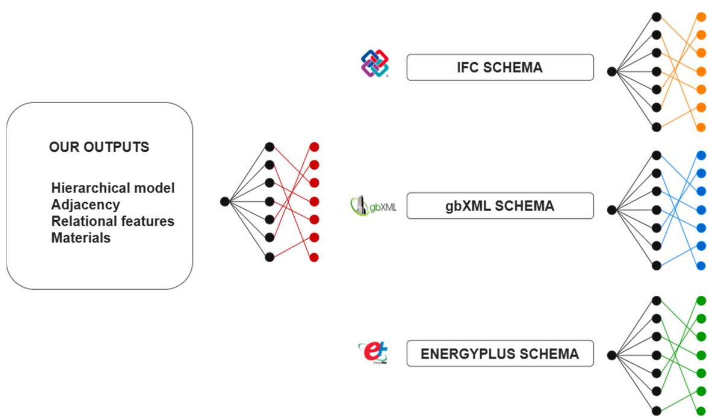


Fig. 52. Schema-based interoperability workflow via parser–transformer–writer (PTW) modules.

The MultiDiGraph (Section 2.10) supports computational and simulation workflows, while the RDF view enables semantic querying and interoperability.

	IFC	gbXML	epJSON
Format file	STEP (EXPRESS)	XML (XSD schema based)	JSON
Encoding	Plain text, line-based	XML tags	JSON with nested keys
Structure	Object graph network	Hierarchical Tree	Key-value
Application	BIM coordination	Energy modelling	Energy simulation
Orientation	Geometry-oriented	Space and zone centric	Simulation oriented
Geometry detail	High/medium detail (LOD)	Planar simplified surfaces	Planar geometry, vertex array

Table 13. Comparison of IFC, gbXML and epJSON formats in terms of file structure, encoding, application focus and geometric representation.

This dual representation enables efficient computation, semantic interoperability, and dynamic updates. Building on this structure, three Parser–Transformer–Writer (PTW) modules are implemented to address interoperability requirements and generate:

(i) epJSON for direct ingestion by EnergyPlus (U.S. Department of Energy, 2025),
(ii) IFC (optional within this workflow), and
(iii) gbXML.

Building on the enriched topological and semantic outputs shown in Fig. 50, PTW modules reorganize the hierarchical structure, adjacency relations and material properties, and rewrite them into domain-specific schemas (Fig. 53): IFC for BIM, gbXML supports thermal-zone–based energy exchange and epJSON enables detailed EnergyPlus simulations in a JSON format.

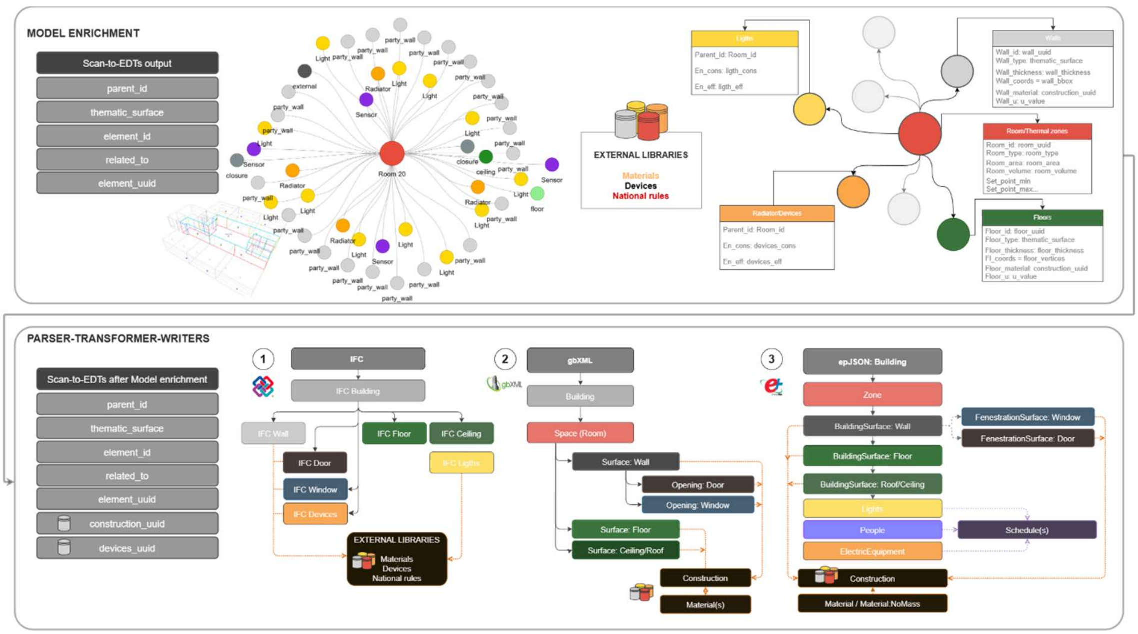


Fig. 53. Top: Scan-to-EDT output attributes and linked graph, illustrating semantic enrichment through external knowledge sources. Bottom: Generation of IFC, gbXML and epJSON models from the graph via the PTW pipeline.

4.7.3.1 PTW1: IFC

The BIM/IFC Parser–Transformer–Writer (PTW) module generates an IFC model starting from the graph representation. First, the parser reads the Scan-to-EDTs CSV output, which contains thematic surface types, parent identifiers, relationships, materials and constructions, and uses this information to reconstruct the hierarchical and topological graph in memory. The building hierarchy is rebuilt from the *parent_id* field, which encodes containment relations, while *thematic_surface* types are used to map custom DT elements to the corresponding IFC classes (Fig. 54).

The transformer then maps DT classes and relationships to IFC 3x4 entities and associations, adding geometry, placements and material assignments. Hierarchical relations become the IFC spatial structure (*Project*, *Site*, *Building*, *Storey*, *Space*), adjacencies define space boundaries and openings, and material and layer data are combined into IFC constructions.

Finally, the writer serializes a valid IFC file, assigning names and *GlobalIds* derived from DT identifiers to maintain traceability. This modular PTW design supports incremental export, schema-level validation and openBIM (https://www.buildingsmart.org/about/openbim/) alignment, while preserving the semantics encoded in the original DT.

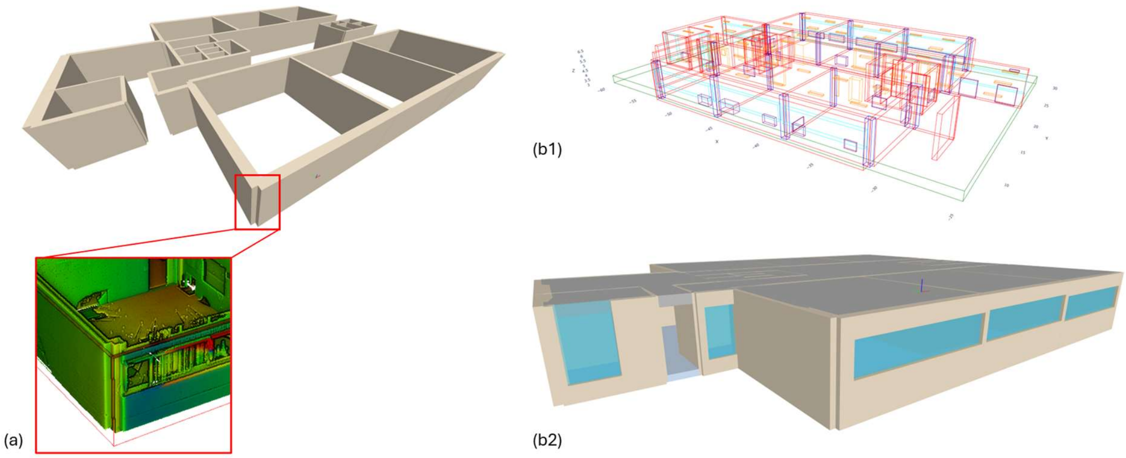


Fig. 54. (a) Reconstructed solid building model with highlighted corner indentation; (b1) solid model overlaid with labeled point cloud and structural wireframe; (b2) solid model with reconstructed openings and façade elements.

4.7.3.2 PTW2: gbXML

This Parser–Transformer–Writer (PTW) module exports analysis-ready gbXML from the graph-based DT, preserving both geometric fidelity and thermal semantics. The gbXML format is an *XML* schema whose structure is close to *Geography Markup Language* (GML), with hierarchical elements, attributes, coordinate lists and explicit spatial relationships (https://www.gbxml.org/) (Fig. 55).

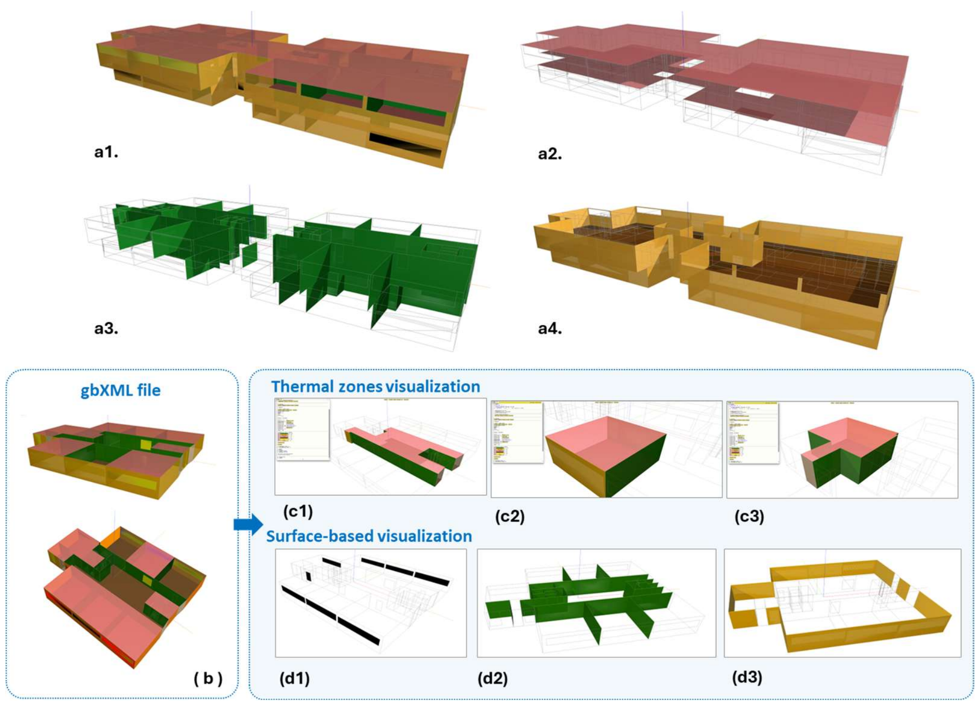


Fig. 55. Multi-level gDT representations (a–b) and their underlying structure exported to gbXML (c) model visualized as thermal zone volumes (c1–c3) and thematic surfaces (d1–d3), demonstrating interoperability with building performance simulation tools.

In the first stage, the parser reads the graph and extracts the spatial hierarchy, surface topology, material assignments, openings and georeferencing information, reconstructing the logical building structure in memory. As in the IFC module, the hierarchy is derived from *parent_id* relationships, while classified surface types define the mapping between graph entities and gbXML elements. The transformer then converts the reconstructed graph into gbXML entities such as Space, Zone, Surface, Opening, Construction and Layer, embedding geometric coordinates, orientation and material properties. Topological relationships are used to determine adjacency, boundary type and exposure conditions, which are critical for energy simulation. Finally, the writer produces a validated gbXML document with stable identifiers inherited from the DT, ensuring compatibility with building energy analysis tools and preserving traceability back to the original graph model.

4.7.3.3 PTW3: epJSON

In recent years, EnergyPlus (U.S. Department of Energy, 2025) has introduced a JSON-based input format (epJSON, https://energyplus.readthedocs.io/en/latest/schema.html) alongside the traditional .idf, enabling more flexible interoperability with external tools. In the proposed workflow, epJSON is produced directly from the graph-based DT using a deterministic mapping process. The mapper walks through the stable identifiers of the DT, spanning building, storey, room or shell, face, edge and vertex, and translates them into EnergyPlus objects in accordance with the Input Data Dictionary (IDD). Topological information, including adjacency and boundary classification, determines the creation of *Zone*, *Surface:HeatTransfer*, *ZoneMixing* and related envelope objects.

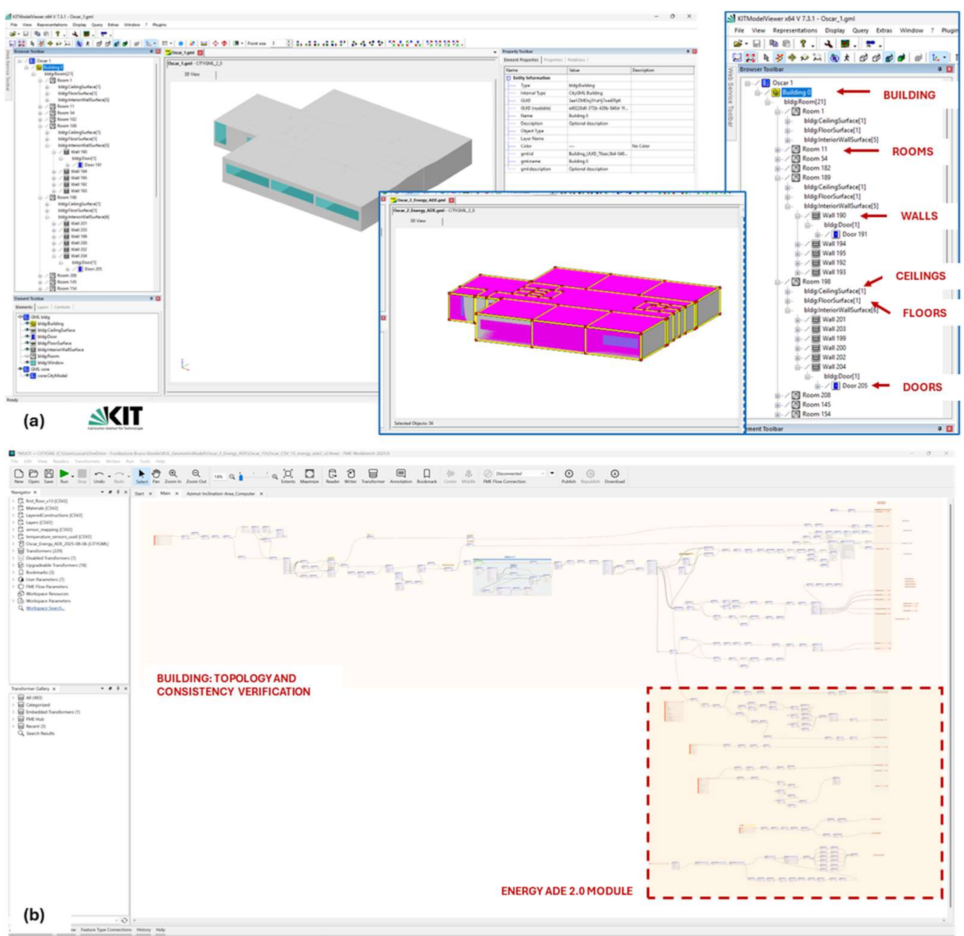


Fig. 56. CityGML export and ADE 2.0 enrichment using FME. (a) CityGML building model visualization and hierarchical object structure (building, rooms, walls, floors, ceilings, doors). (b) FME workspace implementing topology verification and ADE 2.0–based energy enrichment for CityGML interoperability.

Material layers from the construction definitions are converted into Material[*] and Construction[*] entries, with U-values computed from thermal properties. Semantic links to national standards are resolved into schedules, design specifications, setpoint managers and the relevant HVAC objects, resulting in an epJSON file that is fully compliant with the EnergyPlus schema and ready for simulation. A final test explored the generation of a CityGML-compliant model from the proposed workflow outputs. For a subset of the school building, the process produced a CityGML model enriched with Energy ADE 3.0 (Agugiaro and Păsălă, 2025), corresponding to an effective LoD4 representation, as it includes internal spaces and building components. As shown in Figures 56, the model captures the geometric decomposition of the building into floors, rooms, walls, ceilings, and slabs, enabling a detailed semantic and topological organization of indoor spaces.

The hierarchical structure, illustrated in Figure 55 b, explicitly links the building to its rooms and boundary surfaces, facilitating energy-related queries and simulations. Energy ADE attributes associate thermal zones, constructions, and boundary conditions with each interior element, enabling the direct extraction of simulation-ready inputs. This enriched CityGML representation provides a consistent bridge between geometric modeling, semantic structuring, and thermal analysis, supporting downstream CFD simulations and sensor placement strategies within a unified data framework.

4.8 Energy simulations

4.8.1 EnergyPlus

After the transformation in the main format files, the proposed pipeline takes the retrofit specifications as input, compiles them directly into epJSON descriptions and sets up the corresponding simulation cases (Section 2.11). As a first step, the global graph representation is used to reconstruct the as-is configuration of the building, which defines the Baseline scenario (Fig. 57, step 1).

In the retrofit stage (step 2), candidate components, such as external wall segments $\mathcal{W}_{\text{ext}}$ or openings $\mathcal{O}$, are identified through typed queries on the *thematic_surface* table. The DCEL-derived classification (internal vs. external) and persistent UUIDs can be further refined using spatial predicates in PostgreSQL and PostGIS. For each selected component, the procedure gathers its UUID, element category, and construction_id, then follows the associated layer_id and material_id chain in order to compute the U-value [$\mathrm{W\,m^{-2}\,K^{-1}}$].

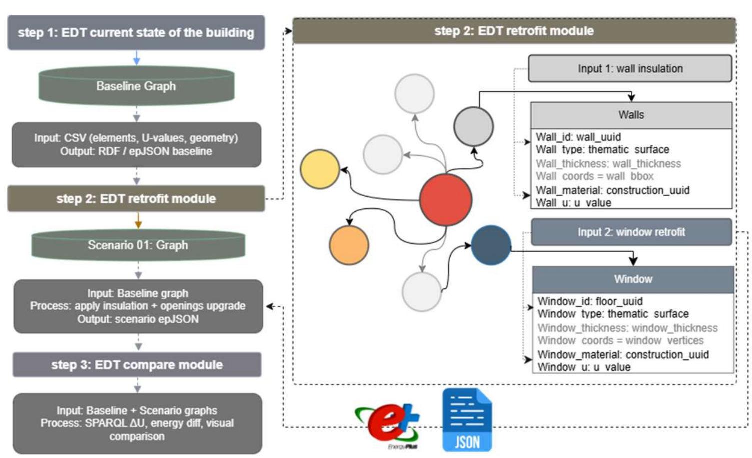


Fig. 57. Modular pipeline for scenario generation. The baseline building state is encoded as a graph and progressively modified through user-driven retrofit inputs, dynamically updating the graph to generate scenario-specific epJSON models for energy analysis.

Currently, two typologies of retrofit actions are implemented:

1. *Thermal upgrading of opaque envelope components* (e.g. walls and roofs) by inserting or reassigning layers.
2. *Improvement of exterior glazing and openings* to meet a prescribed U_{target}.

These changes are executed as transactional operations (e.g. adding or replacing insulation layers, enforcing U_{target}), and the updated layer stacks together with the resulting U-values are written into the scenario-specific epJSON file. Each scenario is then run in EnergyPlus, with inputs and timestamps logged to allow direct comparison against the baseline (step 3).

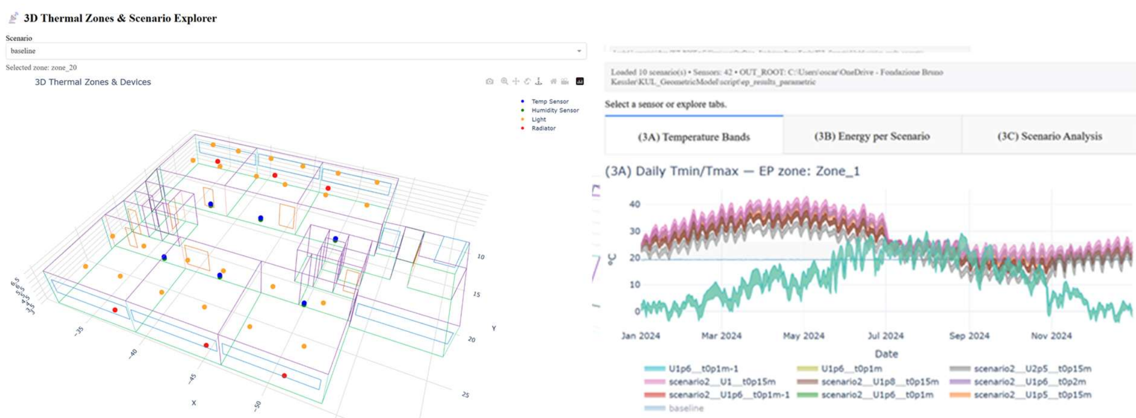


Fig. 58. Visualization of EDT results. Left: 3D navigation and visualization of the building model with sensor locations and semantic elements. Right: interactive analysis of simulation outputs, showing temperature bands and energy performance across multiple retrofit scenarios.

Through this approach, simulation-ready DTL2 instances can be generated on the fly by modifying material and construction assignments for retrofit analysis and performance simulation. In this proof-of-concept implementation, every scenario returns annual energy-use indicators while an API exposes the results for side-by-side comparison.

Figure 57 illustrates the outcomes for the school subset, contrasting the baseline configuration with user-defined retrofit variants, namely added insulation or reduced window U-values. Using dynamic graph operations, user inputs, such as insulation thickness (m) or a target U-value for windows, are converted on demand into epJSON models, executed in EnergyPlus and returned together with timestamps and analytical plots for scenario assessment (Fig. 58). The main goal, consisting on obtaining a compliant model from point cloud to energy simulations, has been reached.

For example, when the user specifies an insulation layer (thickness in meters), the algorithm isolates exterior walls, obtains their surface area and construction_id, resolves the corresponding layer_id and inserts an insulation layer from the material library. It then recomputes the thermal resistance of each layer, updates the overall *U-value* and writes the revised construction back to the epJSON. For window elements, the procedure directly sets the U-value to the requested target and records this modification in the epJSON file.

4.8.2 Energy simulations: CFD

In addition to the EnergyPlus simulations, when real-time or historical data are available, DTs can integrate Computational Fluid Dynamics (CFD) and AI-based simulations to analyze indoor environmental conditions and HVAC performance prior to physical interventions. These simulations support energy optimization, accurate HVAC sizing and sensor placement by leveraging detailed DT geometries and measured demand profiles (Roman et al., 2025a).

This study, conducted within the Horizon Europe InCUBE project, investigates a real-world use case at the Centro Servizi Culturali Santa Chiara in Trento, Italy (Fig. 59). It proposes a data-driven pipeline that integrates 3D surveying, CFD and DT geometries to analyze indoor heat distribution and optimize sensor placement. Benchmark with point cloud and panoramic views are available at https://github.com/3DOM-FBK/InCUBE. By leveraging real-time and historical data, the approach supports accurate HVAC sizing, improves system performance and energy efficiency and prevents the oversizing of technological systems, thereby reducing energy waste.

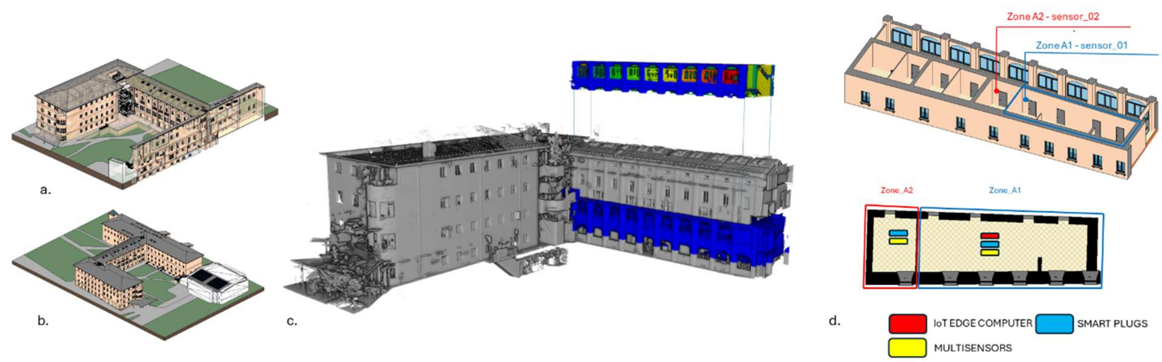


Fig. 59. Santa Chiara building case study and subset selection for DT creation. The figure shows the complete building model, the selected zones and sensor locations used for the DT implementation.

The analyzed area consists of office spaces oriented approximately along a north–south axis. During the ex-ante monitoring phase, IoT devices provided by the InCUBE partner Tera Srl were installed on the first floor, including BEETA Box IoT edge gateways, eleven Z-Wave multiparameter sensors (measuring temperature and relative humidity), thirty-five smart switches, and one CO_2 sensor (Fig. xx).

The study area, referred to as Zone A, comprises two rooms:

- *Zone A1* (*208.65 m³*, three radiators, four windows) and
- *Zone A2* (*56.04 m³*, one radiator, one window).

Continuous measurements of temperature, relative humidity, and CO_2 concentration were recorded at one-minute intervals from 27 February to 6 June 2024 (Fig. 60).

Fig. 60. Indoor sensor deployment and measured environmental data. Left: installed temperature and humidity sensors in the monitored zones. Right: corresponding time series of relative humidity and air temperature used for DT calibration and validation.

This high-resolution dataset enables both aggregated (monthly) and detailed (daily) analyses of indoor environmental conditions and was used to calibrate and evaluate the thermal behavior model within the DT framework.

The DT was enriched with volumetric and semantic information through a topologic graph-based representation, enabling the association of physical properties (Tab. 14), usage patterns (Tab. 15) and boundary conditions (Tab.16) with the reconstructed spaces. This structured model was then converted into computational meshes suitable for CFD simulations.
CFD simulations were carried out to analyze heat transfer and airflow dynamics within the indoor environment, accounting for conduction, convection and internal heat gains. Boundary conditions were defined using external climate data and operational schedules, while radiator positions and characteristics were explicitly included in the model.
The generated meshes were converted to *.stl* format and refined for use in the FEniCS framework, where CFD-based thermal simulations were performed. Within FEniCS, the DT reconstructs the building's structural elements as discretized meshes that define the computational domain. In this context, a mesh represents a partition of the 3D volume into finite elements, commonly tetrahedra in 3D, over which the finite element method (FEM) is applied (Fig. 61).

The mesh preparation workflow involves importing the .obj geometry into Blender (*v. 4.3*), exporting it as *.stl* for compatibility with meshing tools, and subsequently converting the into *.xdmf* format for use in FEniCS simulations. This structured process ensures that the computational domain is properly discretized and optimized for FEM-based heat transfer analysis. Figure 60 illustrates the discretized indoor volume used in the simulations.
For visualization purposes, a grid spacing of v_s = 0.50 m was adopted, while the numerical simulations were conducted at a finer resolution of v_s = 0.25 m, enabling improved accuracy in the representation of thermal gradients.

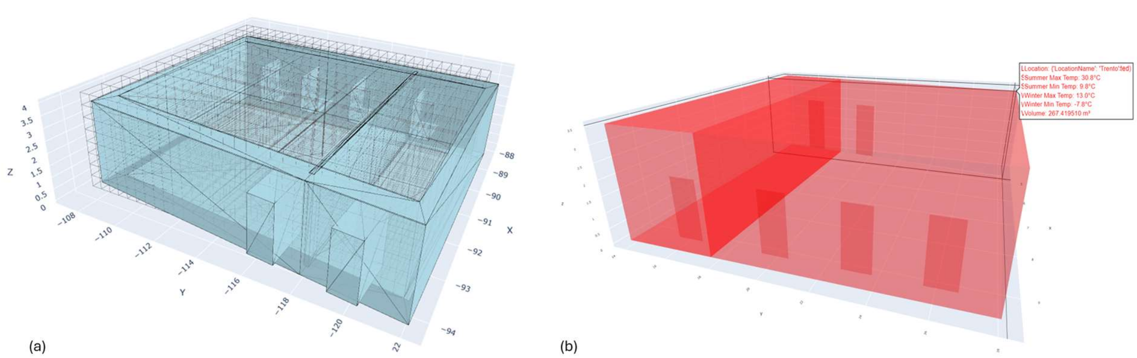


Fig. 61. The geometric model is defined as: (a) solid mesh-based model with discretized volumes; (b) surface-based model with topologic information.

U-values for each building element are computed from layer conductivity and thickness and assigned according to Energy Performance of Buildings Directive (EPBD) guidelines, while heating system parameters follow European standards. Radiators are modelled in their actual positions. The simulations account for convection, conduction and radiation and are calibrated on Zone A using temperature time series from IoT sensors.

Element	U computation	U-value *W/m²K*
External walls	$U_W = \left(\sum_{i=1}^{n} \frac{1}{\lambda_{i,W} \cdot d_{i,W}}\right)^{-1}$	1.25
Windows	$U_{W_{\text{win}}} = \left(\sum_{i=1}^{p} \frac{1}{\lambda_{i,W_{\text{win}}} \cdot d_{i,W_{\text{win}}}}\right)^{-1}$	3.25
Floors	$U_F = \left(\sum_{i=1}^{m} \frac{1}{\lambda_{i,F} \cdot d_{i,F}}\right)^{-1}$	0.80
Ceilings	$U_C = \left(\sum_{i=1}^{m} \frac{1}{\lambda_{i,C} \cdot d_{i,C}}\right)^{-1}$	0.90
Doors	$U_D = \left(\sum_{i=1}^{m} \frac{1}{\lambda_{i,D} \cdot d_{i,D}}\right)^{-1}$	2.50

Tab. 14. Assigned U-values to the structural elements of the Zone A.

Simulated temperature fields were analyzed spatially and temporally to identify areas characterized by low thermal gradients and limited temperature fluctuations. These thermally stable zones are particularly suitable for sensor installation, as they reduce measurement bias caused by localized heat sources or transient effects.

$$occupancy\_schedule(h) = \begin{cases} 0, if\ h \in [19,24) \cup [0,8) \\ 1, if\ h \in [8,10) \\ 2, if\ h \in [10,12) \\ 0, if\ h \in [12,14) \\ 2, if\ h \in [14,19) \\ 0, otherwise \end{cases} \quad \text{(Eq. 21)}$$

Tab. 15. Occupancy schedule definition for the indoor simulations.

To support practical deployment, simulated thermal data were combined with geometric constraints derived from wall surface analyses, excluding unsuitable areas such as windows, doors, and obstructed zones. The integration of CFD outputs with geometric masks allowed the identification of optimal sensor locations that balance physical feasibility and thermal representativeness.

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - \nu\nabla^2\boldsymbol{u} + \nabla p = \boldsymbol{F} \quad \text{(Eq. 22a)}$$

$$\int_\Omega \left(\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - \nu\nabla^2\boldsymbol{u} + \nabla p - \boldsymbol{F}\right) \cdot \boldsymbol{v}\, d\Omega = 0 \quad \text{(Eq. 22b)}$$

$$\frac{\partial T}{\partial t} + \boldsymbol{u} \cdot \nabla T - \alpha\nabla^2 T = S \quad \text{(Eq. 22c)}$$

Where:

- $\boldsymbol{u}$ is the velocity vector field;
- t is time [s, min, h];
- $\frac{\partial \boldsymbol{u}}{\partial t}$ transient term, that is the fluid acceleration;
- $(\boldsymbol{u} \cdot \nabla)\boldsymbol{u}$ is the convective term, the fluid inertia;
- $\nu\nabla^2\boldsymbol{u}$ is the diffusion term, so the viscous effect;
- ∇p is the gradient of pressure;
- $\boldsymbol{F}$, the external forces, when present.

Simulations performed on the vowelized 3D model were calibrated using measured IoT temperature data to ensure consistency between simulated and observed indoor thermal behavior. The indoor volume was discretized into cubic cells with a resolution of $v_s = 0.25\ m$, enabling the evaluation of spatial and temporal temperature variations across the indoor environment throughout the day. This discretization supports the visualization of average daily temperature distributions, particularly in proximity to interior and window-adjacent walls.

Figure 62 illustrates the CFD-based heat distribution within Zone A and highlights two discretized sub-volumes located near window-facing walls at two time steps, $t_1 = 720\ min$ and $t_2 = 1200\ min$. Figures 11a–b show the spatial location of *Volume_01* and *Volume_02* within Zone A, while Figures 11c–f present their corresponding temperature distributions at t_1 and t_2. Simulated temperature fields were quantitatively compared with IoT measurements to assess model accuracy.

The mean absolute error $\bar{E}$ between simulated and measured temperatures was computed as (Eq. 23):

$$\bar{E} = \frac{1}{n}\sum_{i=1}^{n}\left|t_{\mathrm{IoT},i} - t_{\mathrm{simul},i}\right| \quad \text{(Eq. 23)}$$

Where $t_{\mathrm{IoT},i}$ is the temperature recorded by the IoT sensor at time i, $t_{\mathrm{simul},i}$ is the corresponding simulated temperature, and n is the number of samples (e.g., $n = 1440$ for a daily simulation).

The in-house code simulations were conducted on a system equipped with a NVIDIA® GeForce RTX™ 4050 GPU with 6GB GDDR6 memory, ensuring high computational performance and efficiency. Simulations with the in-house FEniCS library took around 10 hours to be computed.

The resulting means are:

Daily error: $$\overline{E_{daily}} = \frac{1}{n}\sum_{i=1}^{n}\left|t_{\mathrm{IoT},i} - t_{\mathrm{simul},i}\right| = 0.644\,°\mathrm{C}, \qquad n = 1440 \quad \text{(Eq. 24a)}$$

Monthy error: $$\overline{E_{monthly}} = \frac{1}{n}\sum_{i=1}^{n}\left|t_{IoT,i} - t_{simul,i}\right| = 1.67\,°C, \qquad n = 43200 \quad \text{(Eq. 24b)}$$

In building energy and CFD-based thermal simulations (Fig. 62), temperature errors below 1–2 °C are widely considered acceptable for real-world applications, given sensor uncertainty, spatial temperature gradients, occupant behavior and modeling simplifications. Errors below 1 °C typically indicate good model calibration, while deviations exceeding 2 °C suggest potential model inconsistencies. In this study, the obtained errors (0.64 °C daily and 1.67 °C monthly) fall within accepted thresholds, confirming the reliability of the calibrated DT–based simulation framework (Fig. 63).

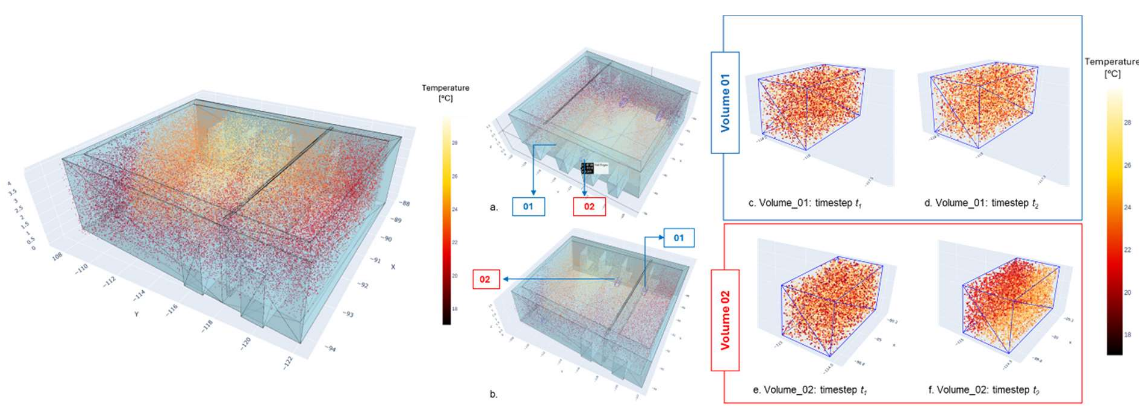


Fig. 62. Volumetric mapping of sensor-derived temperature data within the DT. Left: 3D temperature field visualized across the entire space. Center and right: voxel-based volumetric representations for individual zones, enabling voxel-specific analysis and comparison across different time steps.

The final stage of the workflow focuses on identifying optimal sensor placements. First, orthogonal wall masks are extracted from classified wall elements in the point cloud. Wall surfaces are projected into 2D space, where edge detection and DBSCAN clustering are applied to distinguish available sensor locations from obstructed areas (Fig. 64).

Thermal stagnation regions, where temperatures remain stable across multiple scenarios, are then identified. These zones are ideal for sensor placement, as they ensure reliable measurements with minimal external influence. The resulting masks define the candidate sensor locations.

Daily simulations compute (i) average daily temperatures from hourly data and (ii) monthly averages from daily values. These outputs are integrated into a unified pipeline to determine optimal sensor positions.

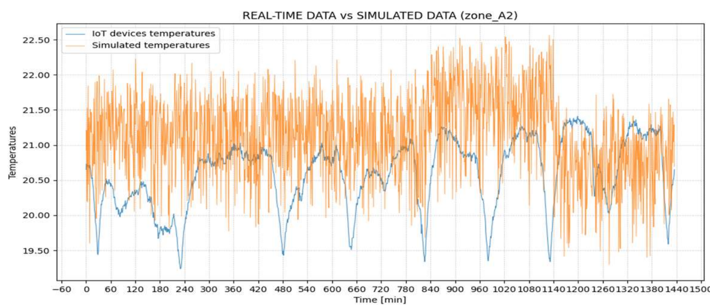


Fig. 63. Comparison between measured (ground-truth) and simulated air temperature for Zone A2. Blue curve represents sensor measurements, while the orange curve shows simulated temperatures over the same time period.

CFD simulations use these temperature values to generate hourly heat distributions, which are converted into point clouds (65a). The workflow proceeds as follows:

1. *Creation of the point cloud from voxel-based simulations:* the simulated temperature in voxels (v_s = 0.25 m) are converted into point cloud (Fig. 65a).
2. *Wall-adjacent point selection:* Points within $d_{\text{wall}} < 1$ m of each wall are extracted (Figure 65b).
3. *Projection and orthoimage generation:* Selected points are projected onto wall planes to produce temperature-annotated orthoimages (Figures 65c–d).
4. *Temporal heat analysis:* Hourly heat distributions are visualized over time (Figure 65e).
5. *Sensor placement determination:* Thermal stagnation zones from CFD simulations, computed using the points temperature of each voxel along the time of simulation, are combined with availability masks to identify optimal sensor locations (Figure 65f).

In particular, sensor placement is determined by combining geometric availability with thermal stagnation derived from point-cloud temperature simulations. Let $\mathcal{P} = \{(x_i, T_i(t))\}$ be the set of wall-adjacent points with temperatures $T_i(t)$ at time t, and let $\pi: \mathbb{R}^3 \rightarrow \Omega \subset \mathbb{Z}^2$ denote the projection onto a wall orthoimage. The temperature of a pixel p at time $t$is computed as

$$T_t(p) = \frac{1}{|\mathcal{P}(p)|} \sum_{i:\, \pi(x_i)=p} T_i\,(t), \qquad \text{(Eq. 25)}$$

where $\mathcal{P}(p)$ is the set of points projected onto p. Thermal stagnation is quantified by the temporal standard deviation

$$\sigma(p) = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(T_t(p) - \mu(p))^2}, \qquad \mu(p) = \frac{1}{N}\sum_{t=1}^{N} T_t\,(p). \tag{Eq. 26}$$

Pixels with $\sigma(p) \leq \tau$ are classified as thermally stable. The final sensor placement mask is defined as

$$M(p) = A(p) \cdot \mathbb{I}[\sigma(p) \leq \tau], \tag{Eq. 27}$$

where $A(p)$ is the availability mask derived from image analysis and $\mathbb{I}[\cdot]$is the indicator function. This formulation ensures that selected sensor locations are both geometrically feasible and thermally stable.
This approach enables targeted sensor placement in thermally stable regions, improving thermal comfort assessment and climate control efficiency.

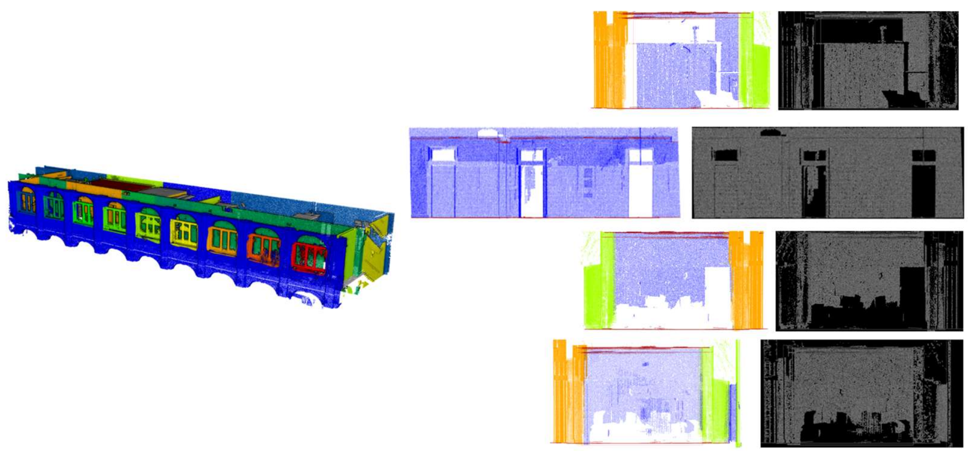


Fig. 64. Left: classified point cloud of the complete floorplan. Right: mask extraction from rasterized wall views, highlighting available areas for sensor placement in white, while unavailable regions are shown in black (grey-scale background).

The proposed semi-automated workflow optimizes sensor placement by converting point cloud data into a 3D model suitable for CFD analysis. Heat diffusion simulations identify stable thermal zones, supporting efficient HVAC sizing and enhanced Building Energy Management Systems (BEMS).

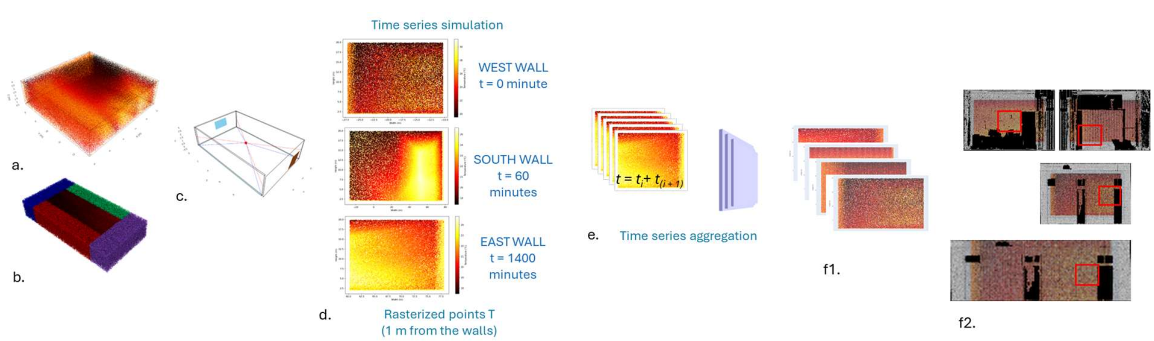


Fig. 65. Sensor placement pipeline: (a) simulated volumetric temperature field; (b) extraction of wall-adjacent points; (c) projection and orthoimage generation of wall surfaces; (d) time-series visualization of wall temperature distributions; (e) temporal aggregation of heat maps; and (f) identification of optimal sensor locations based on thermal stagnation and placement constraints.

Challenges remain in (i) geometric modeling and (ii) simulation accuracy. Automatically generated models introduces mesh complexities that hinder compatibility with CFD tools such as OpenFOAM, requiring further geometric refinement. Simulations slightly overestimate indoor temperatures, likely due to internal heat accumulation, material assumptions, or occupancy effects, while end-of-day temperature drops suggest underestimated envelope heat losses.
The FEniCS library demonstrates high flexibility for geometry handling and heat transfer analysis.
Future work will evaluate the impact of gDT quality and topological accuracy on sensor placement uncertainty, quantifying how geometric representation affects placement reliability as additional data becomes available.

4.9 Results visualization

To visualize baseline simulations, what-if scenarios, and both volumetric and thematic surface models, a web-based interface was developed (HTML) and deployed through a *Dash/Plotly* API. The interface supports interactive 3D visualization and indoor navigation and enables querying of surface classes and construction information. By exposing the underlying graph structure, the API allows users to select thematic surfaces and their topology (e.g., *external walls*, *external openings*, *windows*) and modify associated thermal and physical properties on the fly (Section 4.8.1), thereby enabling rapid scenario definition and exploration.
The dashboard is organized into a 3D viewer and analytical panels (Fig. 66). The left panel renders the reconstructed gbXML geometry on a white background, with boundary surfaces color-coded by thematic class and devices/sensors displayed as clickable markers.

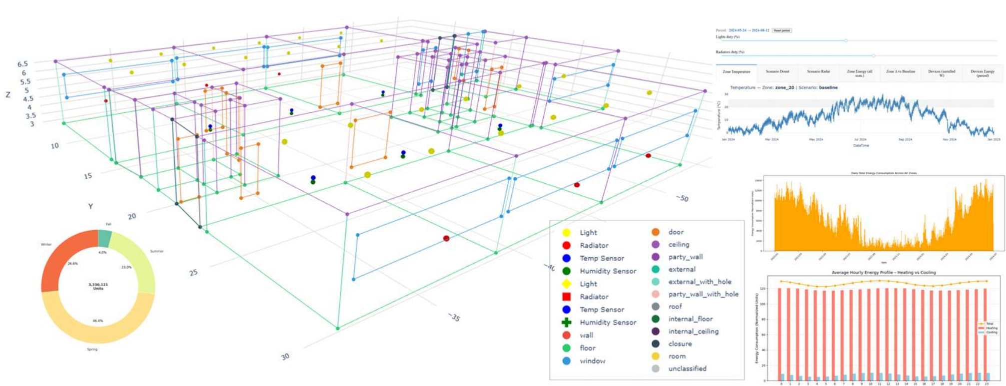


Fig. 66. Application interface for DT visualization and analysis. Left: 3D DT view integrating building elements and sensor locations. Right: interactive dashboards for time-series inspection and scenario-based performance analytics.

Geometry is loaded from a *DataFrame* of WKT polygons with thematic labels and identifiers, while devices are provided as a second *DataFrame* containing (x, y, z) coordinates and device type. When a sensor is selected, it is mapped to a thermal zone using point-in-polygon tests against projected floor polygons, with a nearest-centroid fallback when required.
The right panel then retrieves per-scenario EnergyPlus outputs ($eplus\_out.csv$) for the mapped zone: the *"Temperature"* tab computes daily $T_{\min}$ and $T_{\max}$ from the zone air-temperature time series and plots scenario-wise temperature bands, while the *"Energy"* tab reports heating and cooling demand per scenario using precomputed summaries ($zones\_all.csv$, $totals\_all.csv$) from the simulation step. Scenario folders are automatically uploaded (or loaded from a scenario mapping JSON), enabling direct, side-by-side comparison of baseline and retrofit variants within the same interface (Fig. 67).

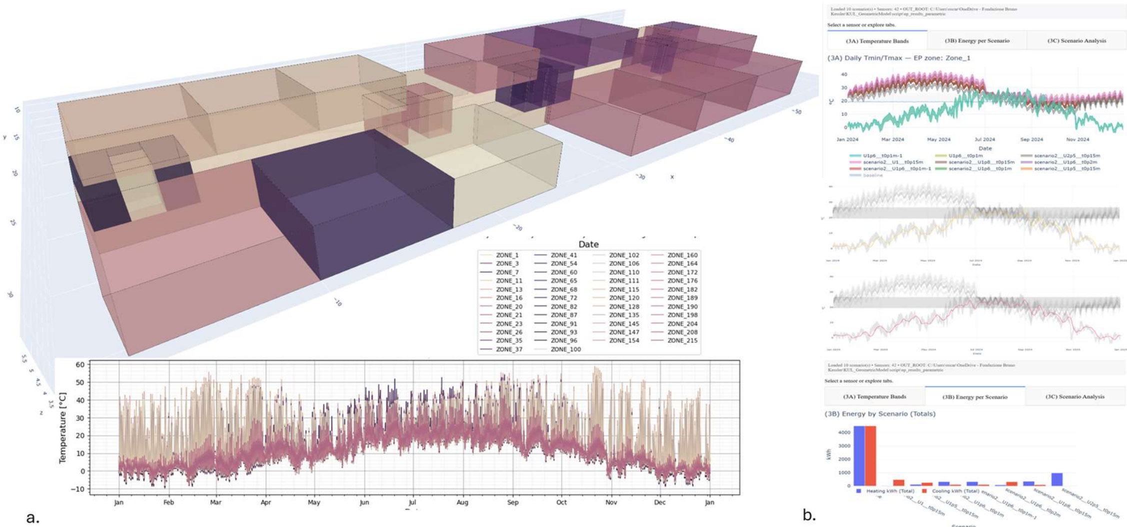


Fig. 67. Integrated DT visualization and analysis. Left: volumetric representation of thermal zones within the building. Bottom: time-series temperature profiles across zones. Right: interactive dashboards showing temperature bands, scenario-based performance and aggregated energy metrics.

5

This chapter presents the results of the proposed Scan-to-Energy Digital Twin (Scan-to-EDT) framework, following the progressive development of the method from raw 3D survey data to simulation-ready energy models. Results are organized according to the main stages of the pipeline, ensuring a clear correspondence between the workflow and its outcomes and enabling a systematic evaluation of performance at each step. The framework is validated on four heterogeneous case studies—residential, institutional, educational, and heritage buildings—captured with different laser scanning technologies and characterized by varying geometric complexity. Quantitative metrics are used to assess semantic classification, geometric reconstruction, topological modeling, and interoperability, while qualitative results illustrate the robustness and consistency of the generated multi-level Digital Twins. Particular emphasis is placed on the integration of volumetric and surface-based representations, graph-based abstractions linking spaces and devices, and standards-compliant exports for energy simulation. Together, these results demonstrate the scalability, robustness, and practical applicability of the proposed approach for Energy Digital Twin generation and analysis.

5. Results

Results are hereafter presented following the progressive development of the method toward an Energy Digital Twin (EDT).
Since the Scan-to-EDTs framework is composed of several sequential stages (Fig. 12, Fig. 23), the results are organized and reported per stage, reflecting individual publications, but largely reporting the results in (Roman et al., 2025c) where the method has been applied to several types of buildings.
This structure ensures a clear correspondence between the workflow and its outcomes and allows for a detailed assessment of performance at each step.
The complete framework has been tested on four case studies (Roman et al., 2025c), in particular (Table 14):
(1) private single-family house from NavVis datasets[1],
(2) a university building in Ghent (KU Leuven campus),
(3) a public school in Italy with two mirrored blocks over two floors, and
(4) an Italian heritage building with irregular and complex geometries.

5.1 Metrics computation

For the evaluation, different metrics are employed, in particular:
- Intersection over Union (IoU) (Eq. 28);
- F1-score (Eq. 29);
- Mean Average Precision for Vertices (mAP_v) (Eq. 33).

These metrics assess the consistency of the proposed Scan-to-EDTs workflow across the main pipeline stages:
- *Classification and instance segmentation:* evaluated using IoU and F1-score.
- *3D reconstruction:* evaluated using IoU and mAP_v.
- *Multi-level DT generation:* evaluated on both volume-based topological models and surface-based models.
- *Interoperability exports (IFC, gbXML* and *epJSON):* evaluated via consistency checks, comparing the number of exported gbXML elements with the corresponding Scan-to-EDTs entities.

Different case studies cover different subsets of the proposed pipeline: some are restricted to Scan-to-BIM reconstruction, while others span the entire workflow, from reconstruction to topological model generation and gbXML export. For clarity, all metrics used to evaluate the different stages and case studies are formally defined in this section.
The metrics used for evaluation are defined as follows.

Classification and instance segmentation

For classification and instance-level evaluation, as wellas for 3D reconstruction steps, Intersection over Union (IoU) and F1-score are employed.
Given a predicted region A and a ground-truth region B (e.g., masks, OBBs or volumetric elements), IoU and F1 are defined as Eq. 28 and Eq. 29,

$$\mathrm{IoU} = \frac{|\ A \cap B\ |}{|\ A \cup B\ |}, \qquad \text{(Eq. 28)}$$

$$F1 = \frac{2\,TP}{2\,TP + FP + FN} \qquad \text{(Eq. 29)}$$

[1] https://www.navvis.com/resources/content-library

Where:
• TP are true positives, predicted OBBs correctly matched to a GT OBB;
• FP are false positives, predicted OBBs not matched to any GT OBB;
• FN are false negatives, GT OBBs not matched by any prediction.
If $\mid A \cup B \mid = 0$, then IoU is set to 0, while if the denominator of F1 is zero (e.g., $TP = FP = FN = 0$, as when there are no predictions and no ground truth), $F1$ is set to 0. The adopted convention is stated alongside the results.
The per-class accuracy is defined as:

$$\mathrm{Acc_c} = \frac{TP_c}{TP_c + FN_c} \quad \text{(Eq. 30a)}$$

$$\mathrm{mAcc} = \frac{1}{\mathrm{C}}\sum_{\mathrm{c}=1}^{\mathrm{C}} \mathrm{Acc_c} \quad \text{(Eq. 30b)}$$

where c indexes classes, C is the number of classes; TPc and FNc are true and false negatives for class c (with positives/negatives defined per class).

$$\mathrm{IoU_c} = \frac{TP_c}{TP_c + FP_c + FN_c} \quad \text{(Eq. 31a)}$$

$$\mathrm{mIoU} = \frac{1}{C}\sum_{c=1}^{C} \mathrm{IoU_c} \quad \text{(Eq. 31b)}$$

Topological models (surfaces and volumes)
For topologic models, agreement is measured using area IoU for surfaces and volume IoU for volumetric cells.
With fixed IoU thresholds $\tau_{\mathrm{iou}}^{\mathrm{surf}}$and $\tau_{\mathrm{iou}}^{\mathrm{vol}}$, precision and recall are given by Eq. 32 and Eq. 33:

$$\mathrm{Precision} = \frac{TP}{TP + FP}, \quad \text{(Eq. 32)}$$

$$\mathrm{Recall} = \frac{TP}{TP + FN}. \quad \text{(Eq. 33)}$$

When predictions include confidence scores, the score threshold is swept to obtain precision–recall (P–R) curves for surfaces and volumes separately; summary statistics may include the maximum F1 over the sweep or area under-curve–style aggregates.

Mean Average Precision for Vertices (mAP$_v$)
For vertex-level evaluation (Fig. 68), a reconstructed vertex is counted as a true positive (TP) at distance threshold η if it lies within Euclidean distance η of any unmatched ground-truth vertex. Otherwise, it is counted as a false positive (FP).
Each ground-truth (GT) vertex can be matched at most once.
For a set of thresholds $\mathcal{T}$and a given $\eta \in \mathcal{T}$, reconstructed geometry vertices are sorted by confidence, a precision–recall curve $P_\eta(R)$ is traced by sweeping the score threshold, and the average precision is computed as (Eq. 34):

$$\mathrm{AP}_v(\eta) = \int_0^1 P_\eta(R)\, dR \quad \text{(Eq. 34)}$$

So, the final mAP_v metric is (Eq. 35):

$$\mathrm{mAP}_v = \frac{1}{|\mathcal{T}|} \sum_{\eta \in \mathcal{T}} \mathrm{AP}_v(\eta) \quad \text{(Eq. 35)}$$

Reconstructed geometries are matched one-to-one to ground truth (GT) OBBs using IoU thresholds $\tau \in$ 0.05, 0.10, 0.15. A prediction with IoU $\geq \tau$ to an unmatched GT is counted as a TP; unmatched predictions are FP and unmatched GT instances are FN.

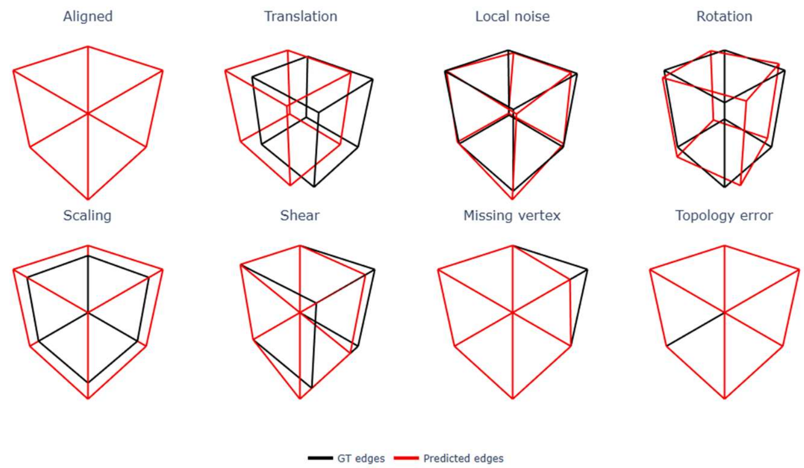


Fig. 68. Examples of geometric and topological deviations used in the evaluation. Ground-truth edges (black) and reconstructed edges (red) illustrate alignment errors captured by the proposed metrics.

5.2 Results overview

The following sections report the results achieved toward the generation of the final multi-level Energy Digital Twin, primarily building on (Roman et al., 2025c). They reflect the consolidated implementation of the Topologic Map framework introduced in (Roman et al., 2024a) and extended in Roman et al. (2024b), together with the simulation workflows reported in (Roman et al., 2025a), and ongoing EnergyPlus-based developments.

Results are presented along three main dimensions. First, the Scan-to-BIM workflow based on deep-learning classification and instance segmentation (Sections 4.2–4.3) generates solid, B-Rep–based models used as input for subsequent stages. Second, the multi-level Digital Twin is established by combining volumetric and surface-based representations with an associated graph structure that manages geometry, topology, and hierarchy. Third, the graph is exploited for semantic enrichment (e.g., materials and systems), standards-based interoperability (IFC, gbXML, epJSON), and simulation-driven what-if scenario execution.

Accordingly, the following sections present results aggregated across all datasets (Section 5.3), followed by a reconstruction and typology-specific analysis (Section 5.4) and a detailed discussion of graph-based representations and simulation results.

5.3 Complete framework: Scan-to-Energy Digital Twin (EDTs)

As explained in Chapter 4, the Scan-to-EDT framework is an end-to-end pipeline that transforms raw three-dimensional survey data into a Digital Twin ready for energy simulation. The evaluation of geometric accuracy and semantic consistency is carried out across five interconnected phases of the pipeline:

• *Geometric reconstruction* (Scan-to-BIM or Scan-to-Mesh), including classification, instance segmentation and solid model creation.

• *Topologic model creation*, including floor-plan extraction, simplified 3D volume reconstruction, surface-based B-Rep generation and graph construction for connectivity and hierarchy.

• *Semantic enrichment,* adding materials and rule-based national code requirements.

• *Interoperability modules,* enabling export to IFC, gbXML, and epJSON.

• *Simulation model generation,* producing an EnergyPlus-ready model and generating the final epJSON input for simulation.

In the following section results are reported with reference to the metrics reported in Section 5.1.

5.3.1 Dataset used

As anticipated at the beginning of the chapter, to test the Scan-to-EDTs framework, the complete pipeline has been applied to four different building typologies acquired with various laser scanning technologies (Table 14):

- *Private single-family house,* taken from both open-access NavVis libraries and dedicated surveys carried out with a mobile laser scanner (MLS).
- *A public university building* in Ghent on the KU Leuven campus, representing a typical European-style institutional building
- *A public school in Italy*, comprising two floorplans and two symmetrical blocks, representative of Italian architectural style
- *A heritage building in Italy*, where the main challenge was handling irregular shapes and complex geometries.

The datasets were acquired using both Mobile Laser Scanning (MLS) and Terrestrial Laser Scanning (TLS), with comparable point cloud densities ranging between 5 mm and 6 mm. The number of available floorplans varies from one to four per site. While *House 01* and *Block N* do not include indoor sensor data, *PEA School* and *Santa Chiara* provide additional measurements, enabling the validation of device to space mapping and energy-oriented reasoning.

Overall, this combination of building typologies, acquisition modalities, and sensor availability provides a diverse experimental setting to assess the robustness and generality of the proposed geometric Digital Twin framework under realistic and heterogeneous conditions.

	House 01	Block N	PEA School	PEA school subset	Santa Chiara
Type of device	MLS	TLS	TLS	TLS	TLS
Density	6 mm	5mm	5 mm	5 mm	5mm
Number of Floorplans	2	4	2	1	1
Contains indoor sensor data	✗	✗	✓	✓	✓

Table 16. Summary of datasets and their features.

5.3.2 Classification phase

Semantic classification (Fig. 69) is carried out by segmenting the point clouds using Point Transformer v3, relying on an internally pretrained model (Roman et al., 2024). The network processes three-dimensional

point coordinates, RGB information when available, and point normals as input features, enabling robust semantic discrimination across different building elements.
For larger datasets, such as the *PEA school* and university building *Block N*, point clouds are partitioned into spatial subsets to improve computational efficiency and memory usage while preserving local geometric continuity. In parallel, 2D image-based detection is employed to complement the three-dimensional segmentation results. In particular, YOLOv8 (Varghese and Sambath, 2024) is used for column detection, while GroundingDINO (Liu et al., 2023) is applied to detect openings such as doors and windows, as described in Section 4.2.2.
This multimodal strategy improves the reliability of semantic labeling by combining three-dimensional geometric cues with two-dimensional visual information.

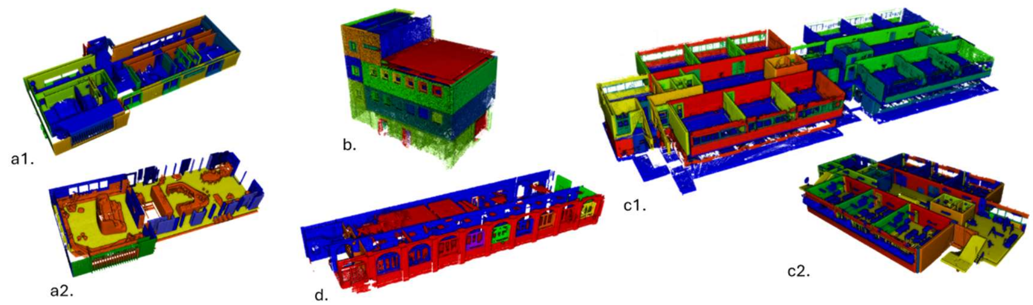


Fig. 69. Classified datasets used in this work: (a1) House 01 ground floor; (a2) House 01 first floor; (b) Campus Ghent, Block N; (c1) PEA School; (c2) PEA School subset; (d) Santa Chiara central wing, first floor.

The two outputs (2D and 3D-based) are then fused to ensure consistency in the 3D reconstruction workflow. As shown in Table 15, major structural elements such as floors, ceilings and walls achieve high accuracy (mIoU > 85%, mAcc > 89%), while columns, doors and windows perform lower (mIoU = 44.1%, 51.6%, 58.8%) due to their smaller size, occlusions and class imbalance. Late fusion improves window reliability and overall accuracy remains sufficient for robust topology reconstruction and energy-zone definition.

	Point Transformer v3 – YOLOv8 – GroundingDINO		
Class	***Std dev [%]***	***mIoU [%]***	***mAcc [%]***
Unclassified	*5.12*	*78.28*	*85.38*
Foors	*2.64*	*94.22*	*96.84*
Ceilings	*2.69*	*92.73*	*95.63*
Walls	*2.45*	*87.51*	*89.42*
Columns	*0.72*	*44.14*	*71.52*
Doors	*1.13*	*51.62*	*65.33*
Windows	*1.56*	*58.83*	*70.24*

Table 17. Classification results obtained using Point Transformer v3, YOLOv8 and GroundingDINO.

5.3.3 Scan-to-BIM reconstruction

The following section presents the results of the Scan-to-BIM workflow for the generation of solid models. A per-class reconstruction strategy (Section 4.2.5) is adopted, whereby each semantic class is reconstructed independently. This approach enables the use of class specific geometric assumptions, such as axis alignment for walls and columns and vertical alignment along the z-axis for openings, leading to improved reconstruction precision and completeness.
Performance is evaluated using the mean Intersection over Union (mIoU) and F1 score, computed from the spatial overlap between predicted elements and minimum Oriented Bounding Boxes (mOBBs) derived from a manually segmented and classified point cloud used as instance level ground truth. Per class evaluation results are summarized in Table 16 and visualized in Figure 70.

The quantitative results reported in Table 16 show a generally consistent performance of the Scan-to-BIM workflow across the different case studies and building typologies, with mIoU values generally ranging between approximately 69% and 96% and F1 scores between 70% and 94%, depending on the semantic class and dataset.

Large planar elements consistently achieve higher accuracy, with mIoU often exceeding 85% and F1 scores above 86%, while more complex or smaller elements show slightly reduced values, reflecting their greater geometric variability and sensitivity to segmentation errors. Variations between floors within the same building, as observed for *House 01*, suggest that scene layout and acquisition conditions have a measurable impact on reconstruction accuracy. The *School PEA* and *Santa Chiara* datasets exhibit strong and balanced results across classes, including in complex and heritage environments, confirming the robustness and generalization capability of the per-class reconstruction strategy across different sensing modalities and architectural typologies.

Elements	Metrics	Columns [%]	Doors [%]	Walls [%]	Windows [%]
House 01 – First floor NavVis dataset	mIoU	–	70.2	88.6	68.1
	F1	–	72.5	85.2	70.3
House 01 – Ground Floor NavVis dataset	mIoU	85.2	68.9	89.9	67.8
	F1	86.9	70.7	87.0	69.4
Block N KULeuven Campus	mIoU	–	73.4	91.7	71.3
	F1	–	75.1	88.6	72.8
School PEA subset Italy	mIoU	78.3	76.4	96.3	75.8
	F1	81.2	77.9	94.0	76.7
School PEA Italy	mIoU	85.2	69.6	89.3	68.0
	F1	86.9	71.6	86.1	69.9
Santa Chiara Trento, Italy	mIoU	75.1	77.3	94.8	76.2
	F1	89.1	78.2	92.0	77.5

Table 18. Quantitative evaluation of the Scan-to-BIM workflow using mIoU and F1 score for different building types and datasets.

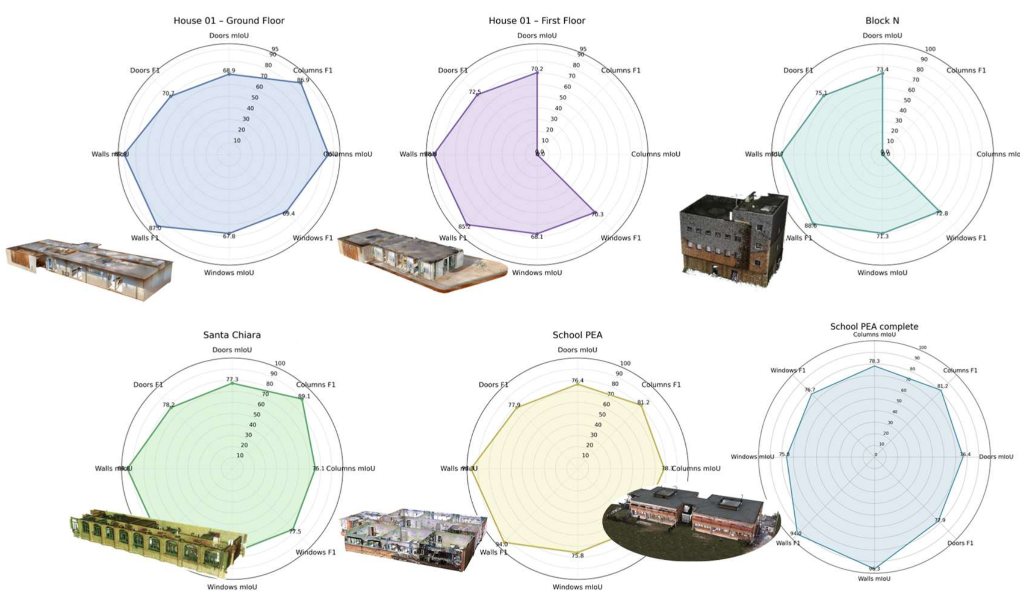


Fig. 70. Metrics results for reconstructed solid-based elements reported per-classes.

5.3.4 Topologic solid model (TSM) & Topologic surface model

The Topologic Solid Model (TSM) and B-Rep reconstruction results are illustrated in Figure 70.
The quantitative evaluation, reported in Table 17 and Figure 70, demonstrates that TSMs, also referred to as *CellComplex* models, consistently outperform B=Rep models across all evaluated metrics.
The accuracy of B-Rep surface objects is assessed against manually generated ground truth (GT) models, with particular emphasis on the mean Intersection over Union (mIoU). As shown in Figure 70, the evaluation is based on oriented bounding boxes computed for wall elements only, using a set of average distance thresholds ranging from 0.05 m to 0.15 m to determine spatial overlap. This evaluation protocol highlights the higher geometric consistency and volumetric completeness achieved by the TSM representation compared to surface-based B-Rep models.
Both representations achieve high precision, with values between 0.93 and 0.98 for TSMs and 0.74 and 0.95 for B-Rep surface-based models, indicating good geometric accuracy when overlaps occur. However, surface-based models exhibit substantially lower recall, ranging from 0.57 to 0.90, compared to 0.87 to 0.96 for TSMs. This gap directly impacts the overall performance, resulting in lower F1 scores (0.65 to 0.89 for surface models versus 0.90 to 0.97 for TSM) and reduced IoU values (0.72 to 0.95 for B-Rep versus 0.93 to 0.98 for TSM).

	Topologic Solid Model (TSM)					*Topologic Surface Model (B-Rep)*				
Building	***Precision***	***Recall***	***F1***	***mIoU***	***mAPv***	***Precision***	***Recall***	***F1***	***mIoU***	***mAP)***
House 1	*0.955*	*0.912*	*0.933*	*0.953*	*0.941*	*0.910*	*0.890*	*0.900*	*0.860*	*0.861*
Block N	*0.972*	*0.946*	*0.959*	*0.973*	*0.962*	*0.949*	*0.929*	*0.939*	*0.899*	*0.892*
School	*0.930*	*0.867*	*0.897*	*0.927*	*0.899*	*0.930*	*0.867*	*0.897*	*0.927*	*0.890*
School subset	0.930	0.867	0.897	0.927	-	0.887	0.867	0.877	0.837	-
Heritage building	*0.937*	*0.879*	*0.907*	*0.934*	*0.921*	*0.898*	*0.878*	*0.888*	*0.848*	*0.857*

Table 19. Topologic solid and surface-based model metrics reported as average value for *th* = 0.05 m, 0.10 m, 0.15 m.

These differences reflect the underlying modeling strategies and, mostly, the highest complexity required to produce the surface-based models.
TSMs explicitly encode enclosed volumetric elements, which align more naturally with the GT wall definitions, whereas B-Rep models rely on surface representations derived from wall centerlines.
The advantage of the TSM approach becomes particularly evident in complex scenarios, such as the *PEA School* and *Santa Chiara* case studies, where dense wall layouts, high number of partitions and irregularities especially in the external shape and in the heritage geometries challenge surface-based reconstruction and DCEL completion.
Despite these discrepancies, B-Rep models remain adequate for energy performance applications, as they preserve dimensions, structural logic, semantic clarity, and topological integrity, and have been visually verified in FME® (FME software) for consistency and correctness. The workflow ultimately produces a multi-level DT in which TSM and B-Rep models coexist and are linked to the graph node structure, and the resulting gDT can be exported to gbXML for interoperable use in building performance simulation tools.

5.4 Results for building typology

This section presents the reconstruction results for the different building typologies considered in the study, focusing on the main steps of the three-dimensional reconstruction process.
In particular, it details how semantic segmentation, class specific geometric modeling, and topologic solid generation contribute to the final DT representations across residential, educational and heritage buildings.

- ***House 01 – NavVis dataset***

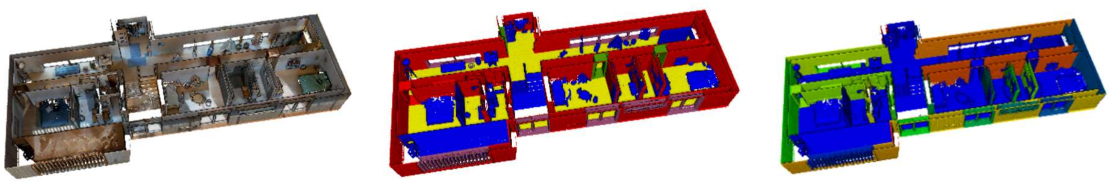
House 01 stands as a showcase of Bau Fritz's commitment to sustainable, modern living, an elegant fusion of eco-friendly materials, precision engineering, and contemporary design. Captured with NavVis VLX (https://www.navvis.com/) and transformed into a high-quality BIM model, this home highlights the company's innovative approach to creating healthy, energy efficient living spaces built for the future.

	House 01
Precision	*6 mm*
Size	*398 m^2*
# Rooms	*9*
Scanning time	*40 min*

Table 20. House 1 dataset information.

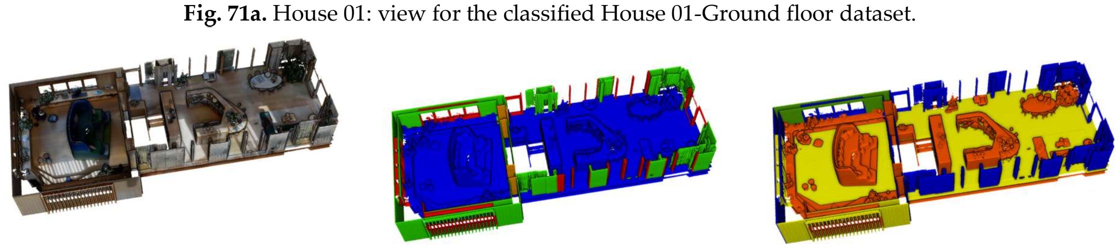
Classification & Instance segmentation phase

Table 18 summarizes the features of the dataset, which was used directly without any preprocessing, apart from the initial extraction of the building itself from the surrounding environment. Figure 71 illustrates the results of the classification and instance segmentation phase: (a) shows the input dataset of the ground floor, (b) the *class_id* classification, and (c) the instance segmentation with *objects_id*. The metrics for the two subsets are reported in Figure 71a-71b and Table 15.



(a) (b) (c)

Fig. 71a. House 01: view for the classified House 01-Ground floor dataset.



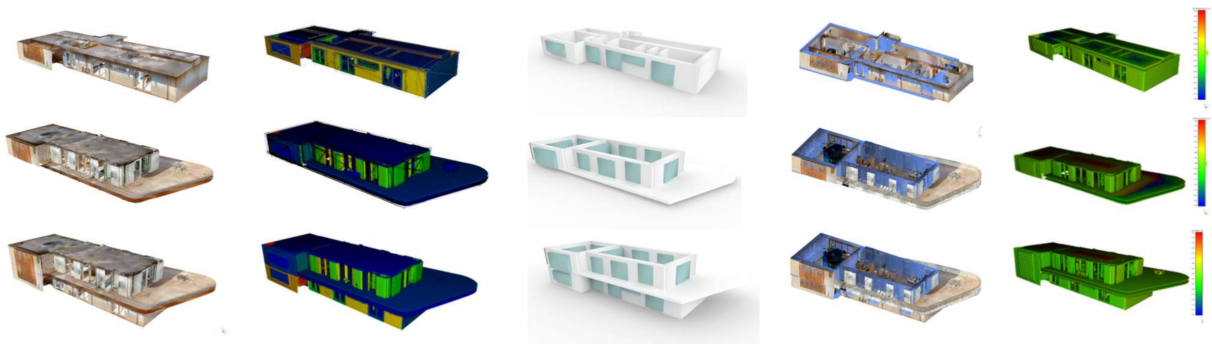
(a) (b) (c)

Fig. 71b. House 01: view for the classified House 01-First floor dataset.

Reconstruction phase

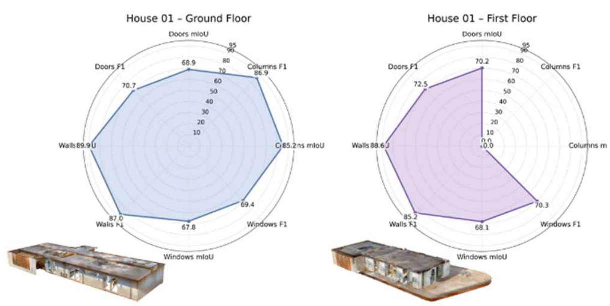
The reconstruction phase, as described in Section 4.2, is performed separately for each class. Figure 72 graphically illustrates: (a) the input point cloud, for a single floor and for the entire building, (b) the classified point cloud, (c) the simplified 3D solid model, (d) the IFC wall reconstruction, and (e) the final cloud-to-mesh (C2M) comparison. Metrics per class are reported in Table 16.

The reconstruction results demonstrate strong overall performance across the main architectural classes, with particularly high accuracy in wall and column reconstruction. Walls achieved the highest mIoU values (89.9% and 88.6% across the two floors), reflecting the robustness of the pipeline in capturing large, continuous structural elements. Columns also show high performances, with an mIoU of 85.2% and an F1 score of 86.9%, although results are available for only one floor due to their limited presence in the dataset. Openings such as doors and windows show moderately lower accuracy compared to structural elements, with mIoU values in the 68–70 percent range and F1 scores in the low 70s, reflecting the greater geometric variability and smaller size of these components. The framework performs very well on structural elements, while smaller and more geometrically complex components remain more challenging.



(a) (b) (c) (d) (e)

Fig. 72. House 01: Scan-to-BIM reconstruction process: (a) Raw point cloud; (b) Instance-segmented point cloud; (c) Generated solid model output; (d) IFC wall-class reconstruction results; (e) Evaluation metrics comparing the solid model to the input point cloud.

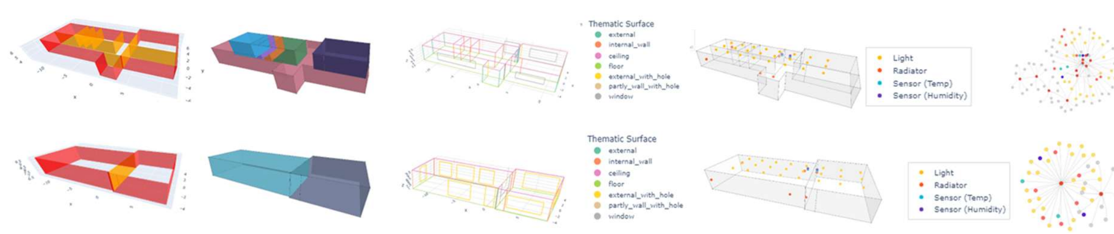


		House 01 – Ground Floor	*House 01 – First Floor*
	mIoU	85.2	-
Columns	*F1*	86.9	-
	mIoU	68.9	70.2
Doors	*F1*	70.7	72.5
	mIoU	89.9	88.6
Walls	*F1*	87.0	85.2
	mIoU	67.8	68.1
Windows	*F1*	69.4	70.3

Table 21. House 01: Metrics results for solid-based elements reported for the per-class reconstruction.

Topologic models

Figure 73 illustrates the House 01 reconstructed multi-level Digital Twin, highlighting the volumetric and surface-based representations together with their semantic and topological organization. The visualizations are generated from custom outputs using Topologicpy for topological modeling and PyVista for three-dimensional rendering, and include thematic surfaces, spatial partitions and the spatial distribution of building elements, energy-related elements and sensors.



(a) (b) (c) (d) (e)

Fig. 73. House 01: (a) topological, internal, external, and closure surfaces; (b) solid model; (c) complete topologic model; (d) topologic model and devices; (e) comprehensive graph of structural and energy-related elements in building.

Interoperability

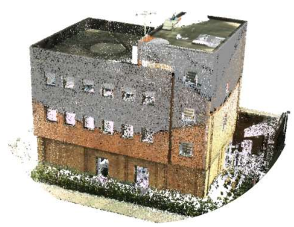
The resulting graph structured representation is processed by the parser transformer writer, which converts the topological and semantic information into a gbXML compliant schema for downstream

energy simulation and interoperability workflows. The generated outputs are shown in Figure 74 and were additionally verified using FME software.
The computational performance of the Scan-to-EDT pipeline is evaluated by analyzing the processing time required for each algorithmic step and its relative contribution to the overall workflow.

Fig. 74. House 01: (a) FEM based representation generated from the parser transformer writer; (b) corresponding gbXML model derived from the graph structured Digital Twin and validated through FME.

- ***Block N, University block***

Block N is a university building located on the KU Leuven Ghent Campus in Belgium.
The survey of the building was analyzed and reported in (Bassier et al., 2019) and the main characteristics of both the survey and the building are summarized in Table 20.

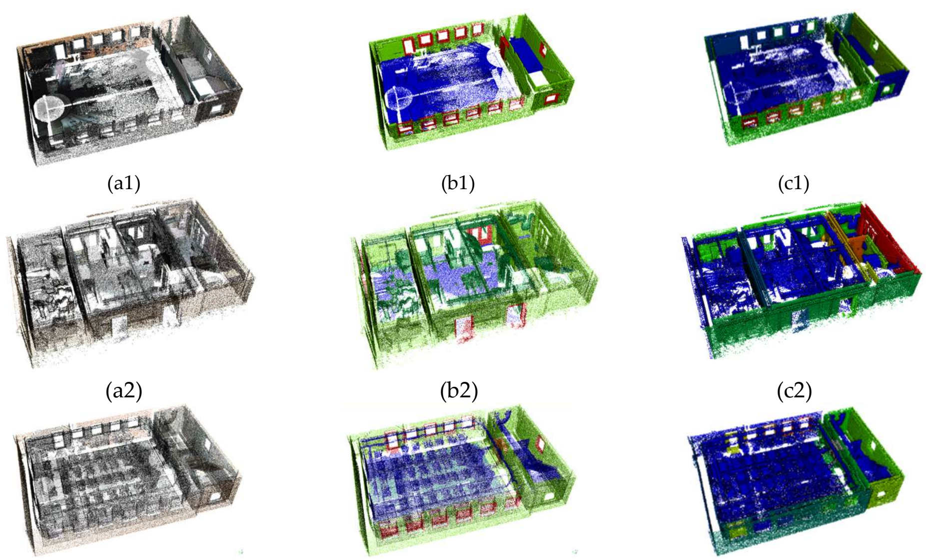


	Block N
Precision	*5 mm*
Size	*650 m^2*
# Rooms	*7*
Scanning time	-

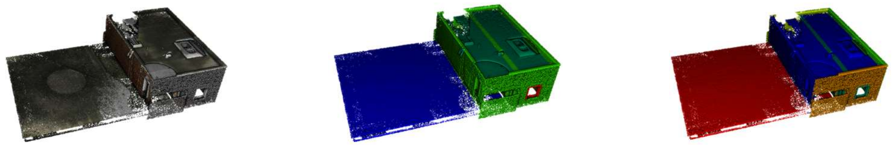
Table 22. Block N dataset information.

Classification & Instance segmentation phase

Table 20 summarizes the dataset, while Fig. 75 shows the per-floor input (a), the semantic classification (b) and the instance segmentation (c). Metrics are reported in Table 15.



(a1) (b1) (c1)
(a2) (b2) (c2)
(a3) (b3) (c3)



(a4) (b4) (c4)

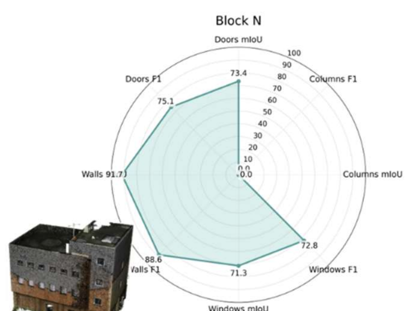
Fig. 75. Block N: view for the classified point cloud.

Reconstruction phase

Figure 76, from left to right, shows the RGB point cloud acquired from scans, the semantically classified point cloud, the reconstructed solid model, the IFC wall reconstruction, and the cloud-to-mesh (C2M) comparison highlighting geometric deviations.

Fig. 76. Block N: Scan-to-BIM reconstruction process: (a) Raw point cloud; (b) Instance-segmented point cloud; (c) Generated solid model output; (d) IFC wall-class reconstruction results; (e) Evaluation metrics comparing the solid model to the input point cloud.

The reconstruction results show, as in the previous dataset, strong performance across the main architectural classes, with walls reconstructed most accurately. Walls reach the highest mIoU (up to 89.9%), confirming the robustness of the pipeline for large, continuous structural elements. Note that Block N does not contain columns, and therefore the column class is not reported for this dataset.



Element	mIoU (%)	F1 (%)
Columns	-	-
Doors	*73.40*	*75.10*
Walls	*91.70*	*88.60*
Windows	*71.30*	*72.80*

Table 23. Block N: metrics results for solid-based elements reported for the per-class reconstruction.

Topologic models

The Topologic model (Fig. 77) is reconstructed independently for each floor plan and subsequently assembled to form the complete multi-storey building.

This approach ensures consistency across levels, and the resulting topology is further verified using FME software to confirm its structural correctness.

Interoperability

As anticipated, the consistency of the topology has been tested and verified using FME, including the application of the PTW workflow to the generated gbXML model. The algorithm demonstrates strong robustness in preserving topological consistency; however, as illustrated in Figure 78, a front-facing opening is not transferred to the gbXML representation.

As a result, the corresponding thematic surface is missing in the exported model.

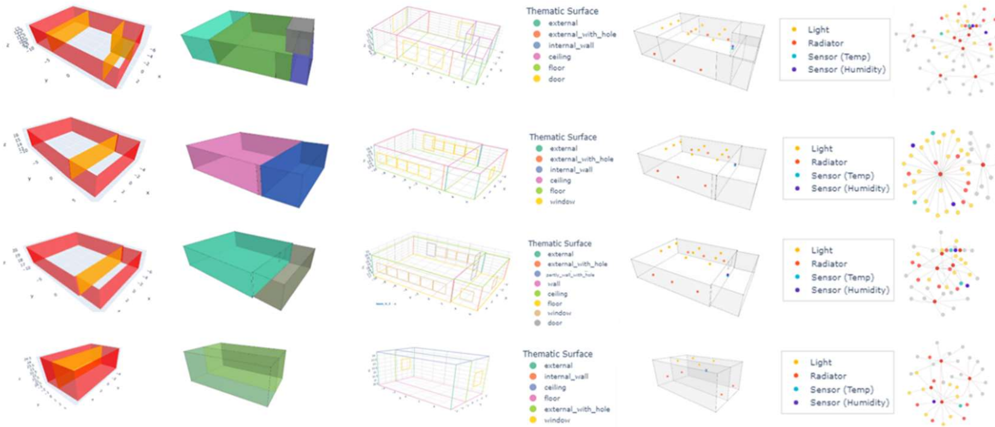


Fig. 77. Block N: (a) topological, internal, external, and closure surfaces; (b) solid model; (c) complete topologic model; (d) topologic model and devices; (e) comprehensive graph of structural and energy-related elements in building.

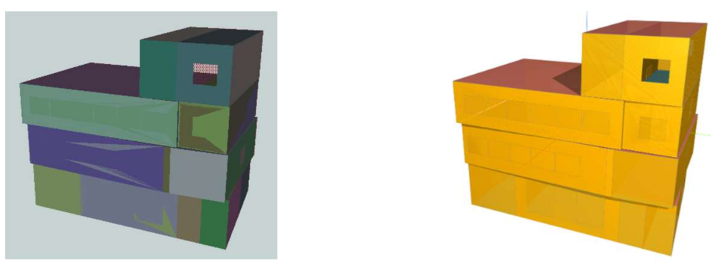


Fig. 78. Block N: (a) FME software-based representation generated from the parser transformer writer; (b) corresponding gbXML model derived from the graph structured Digital Twin and validated through FME.

- ***Santa Chiara – Heritage building***

The Santa Chiara case study represents a complex heritage building in the city of Trento, characterized by irregular geometries, thick masonry walls and fine spatial partitions. These features make it a challenging test case for three-dimensional reconstruction, topological modeling and semantic consistency within the proposed Scan-to-EDT framework.

	Santa Chiara building
Precision	*5 mm*
Size	*480 m²*
# Rooms	4
Scanning time	-

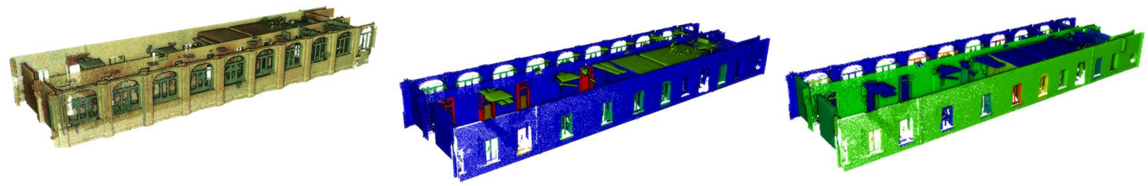
Table 24. Santa Chiara building dataset information.

Classification & Instance segmentation phase

Table 22 summarizes the dataset, while Fig. 79 shows the input point cloud(a), the semantic classification (b) and the instance segmentation (c). Metrics are reported in Table 15.

Reconstruction phase

The reconstruction phase, as described in Section 4.2, is performed separately for each class. Figure 80 graphically illustrates: (a) the input point cloud, for a single floor and for the entire building, (b) the classified point cloud, (c) the simplified 3D solid model, (d) the IFC wall reconstruction, and (e) the final cloud-to-mesh (C2M) comparison. Metrics per class are reported in Table 16.



(a) (b) (c)

Fig. 79. Santa Chiara building: view for the classified point cloud.

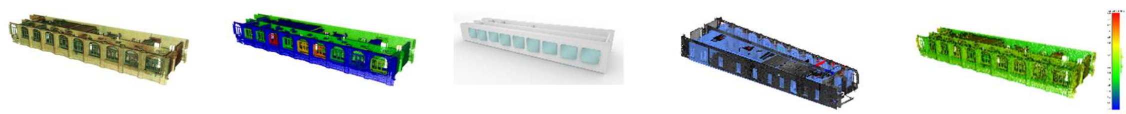
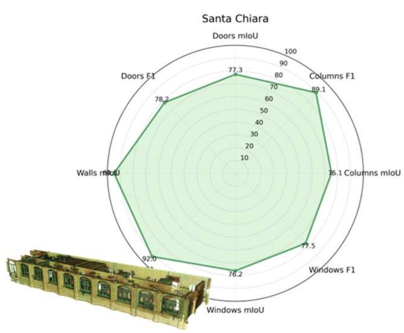


Fig. 80. Santa Chiara building: Scan-to-BIM reconstruction process: (a) Raw point cloud; (b) Instance-segmented point cloud; (c) Generated solid model output; (d) IFC wall-class reconstruction results; (e) Evaluation metrics comparing the solid model to the input point cloud.



	Metrics	*Santa Chiara building*
	mIoU	75.1
Columns	*F1*	89.1
	mIoU	77.3
Doors	*F1*	78.2
	mIoU	94.8
Walls	*F1*	92.0
	mIoU	76.2
Windows	*F1*	77.5

Table 25. Santa Chiara building: metrics results for solid-based elements reported for the per-class reconstruction.

Topologic models

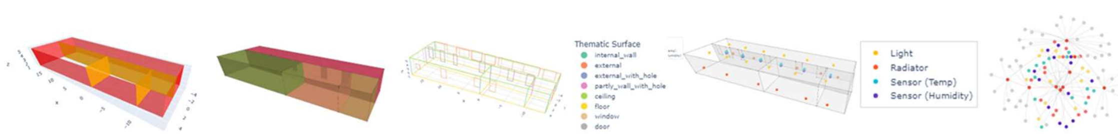


Fig. 81. Santa Chiara building: (a) topological, internal, external, and closure surfaces; (b) solid model; (c) complete topologic model; (d) topologic model and devices; (e) comprehensive graph of structural and energy-related elements in building.

Interoperability

Fig. 82. Santa Chiara building: (a) FME software-based generated from the parser transformer writer; (b) corresponding gbXML model derived from the graph structured Digital Twin and validated through FME.

- ***School PEA***

The survey of the PEA School was carried out using terrestrial laser scanning techniques, covering both interior and exterior spaces. Exterior data acquisition was performed using a phase shift class 1 laser scanner (Leica Geosystems P30), while interior spaces were captured with a next generation terrestrial laser scanner (Leica Geosystems BLK360).
Exterior point clouds were georeferenced using topographic targets with known coordinates, measured through a rigorously adjusted closed control traverse materialized on site.
Target coordinates were acquired using a total station (Leica Geosystems TCRP 1203+), ensuring accurate spatial referencing. Interior point clouds at ground floor entrances were georeferenced using the same target-based approach, while scans acquired within internal rooms were registered using a cloud-to-cloud strategy implemented in Leica Cyclone Register, which allows the estimation and control of alignment residuals.

	School PEA
Precision	*5 mm*
Size	*ca. 2200 m²*
# Rooms	*74*
Scanning time	-

Table 26. School PEA dataset information.

Classification & Instance segmentation phase

Table 23 summarizes the dataset, while Fig. 83 shows the input point cloud(a), the semantic classification (b) and the instance segmentation (c). Metrics are reported in Table 15.

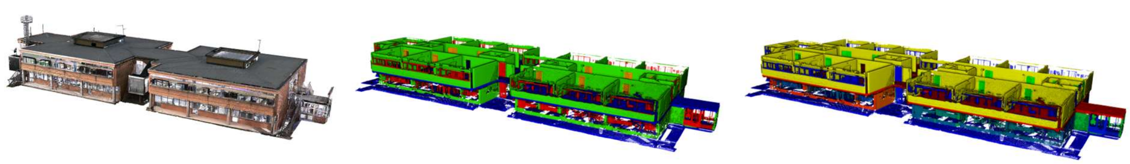
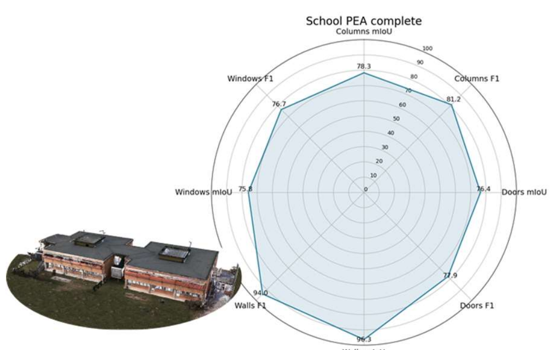


Fig. 83. School PEA: view for the classified point cloud.

Reconstruction phase

As described in Section 4.2, reconstruction is performed independently for each semantic class.
Figure 84 illustrates the main steps, including the input and classified point clouds, the simplified solid model, the IFC wall reconstruction, and the final cloud to mesh comparison, while per class evaluation metrics are reported in Table 16.



	Metrics	*Santa Chiara building*
	mIoU	78.3
Columns	***F1***	81.2
	mIoU	76.4
Doors	***F1***	77.9
	mIoU	96.3
Walls	***F1***	94.0
	mIoU	75.8
Windows	***F1***	76.7

Table 27. School PEA: metrics results for solid-based elements reported for the per-class reconstruction.

Topologic model

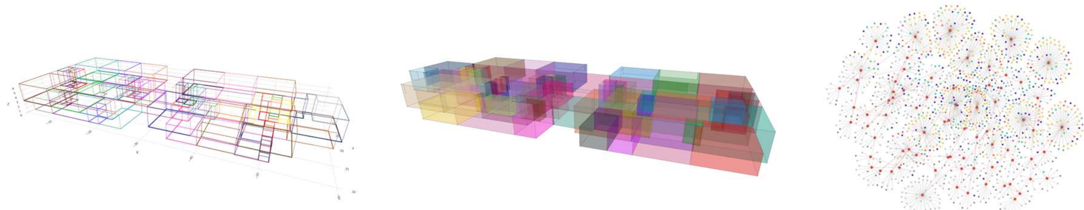


Fig. 84. School PEA: (a) complete topologic model; (b) solid model; (c) comprehensive graph of structural and energy-related elements in building.

Interoperability

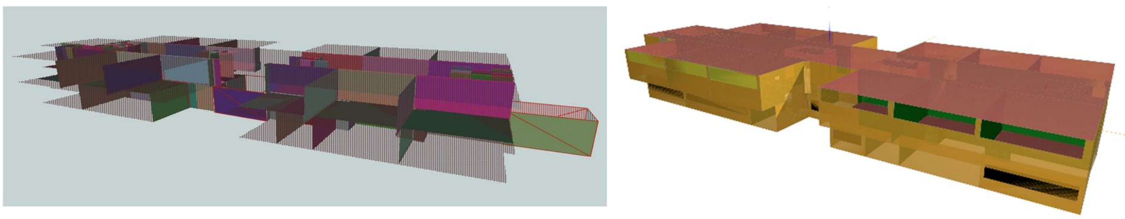


Fig. 85. School PEA: (a) FME software-based generated from the parser transformer writer; (b) corresponding gbXML model derived from the graph structured Digital Twin and validated through FME.

- ***School PEA: subset***

The PEA School subset corresponds to the first floor of the building, selected to provide a focused evaluation area within a larger and more complex dataset. This subset includes classrooms and circulation spaces and is used to assess the reconstruction and DT generation workflow under controlled yet representative indoor conditions.

	School – First floor
Precision	*5 mm*
Size	*525 m²*
# Rooms	*21*
Scanning time	-

Table 28. School PEA subset dataset information.

Classification & Instance segmentation phase

Table 24 summarizes the dataset, while Fig. 86 shows the input point cloud(a), the semantic classification (b) and the instance segmentation (c). Metrics are reported in Table 15.

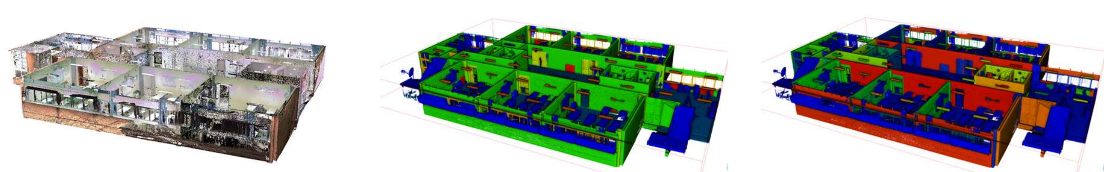
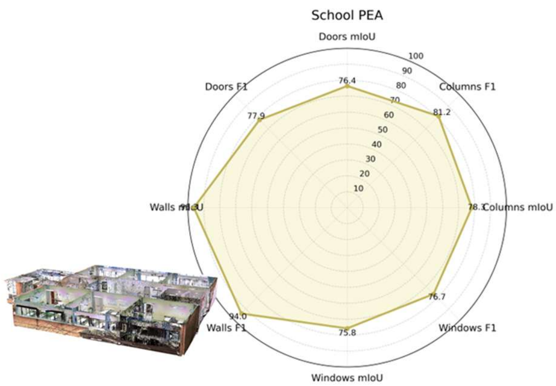


Fig. 86. School PEA subset: view for the classified point cloud.

Reconstruction phase

Fig. 87. School PEA subset: Scan-to-BIM reconstruction process: (a) Raw point cloud; (b) Instance-segmented point cloud; (c) Generated solid model output; (d) IFC wall-class reconstruction results; (e) Evaluation metrics comparing the solid model to the input point cloud.



	Metrics	*Santa Chiara building*
Columns	*mIoU*	78.3
	F1	81.2
Doors	*mIoU*	76.4
	F1	77.9
Walls	*mIoU*	96.3
	F1	94.0
Windows	*mIoU*	75.8
	F1	76.7

Table 29. School PEA subset: metrics results for solid-based elements reported for the per-class reconstruction.

Topologic model

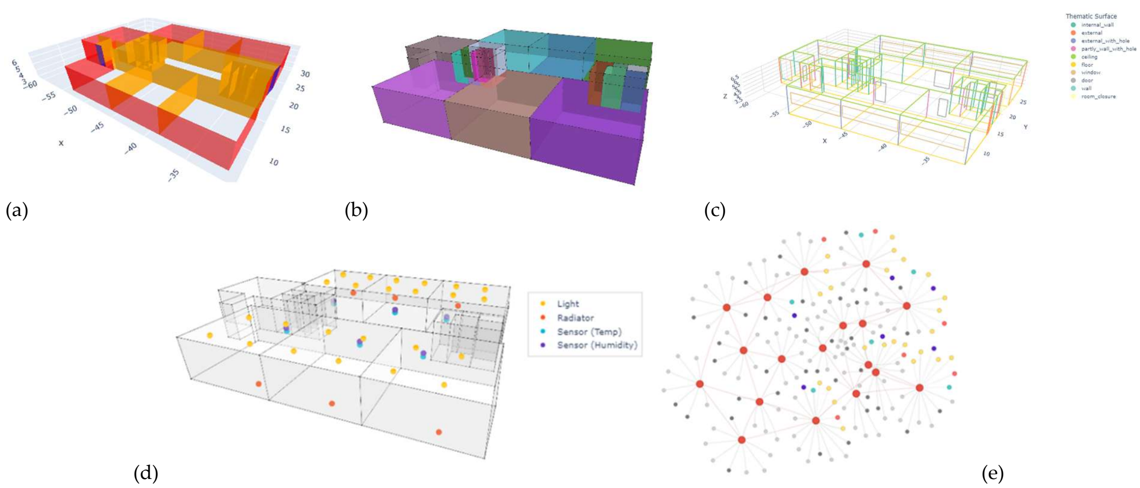


Fig. 88. School PEA subset: (a) topological, internal, external, and closure surfaces; (b) solid model; (c) complete topologic model; (d) topologic model and devices; (e) comprehensive graph of structural and energy-related elements in building.

Interoperability

Topological consistency was verified using FME, including the application of the parser transformer writer workflow to the generated gbXML model.

5.5 Property-Graph-Based Representation and Results

5.5.1 Graph construction and structural properties

A graph-based representation is derived from the reconstructed Energy DT output to explicitly encode spatial, functional and relational dependencies between rooms and building devices. Room volumes are represented as nodes located at cell centroids, while lighting fixtures, radiators and environmental sensors

are modelled as device nodes. Edges encode proximity-based functional relationships, resulting in a multi-layer spatial graph tightly coupled with the underlying geometric model.
Graph construction is fully deterministic and reproducible. Each device is connected to the nearest room centroid, enforcing a unique device–room assignment. Optional room–room edges, defined via k-nearest-neighbour connectivity, capture spatial adjacency and ensure a connected room graph suitable for propagation and aggregation tasks (Fig. 90).

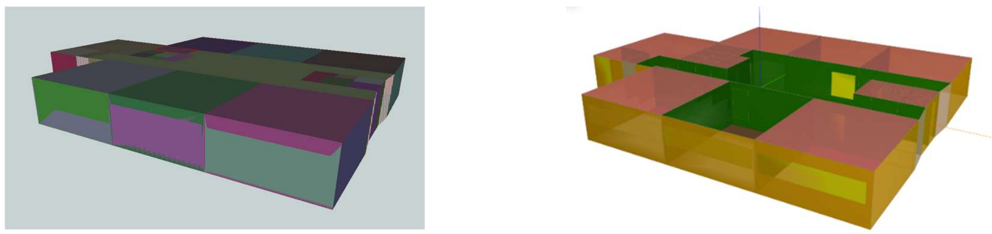


Fig. 89. School PEA subset: (a) FME software-based representation generated from the parser transformer writer; (b) corresponding gbXML model derived from the graph structured Digital Twin and validated through FME.

5.5.2 Device–Room association and topology graph

The device–room association graph shows stable and robust behaviour across all buildings. Devices are repositioned within room volumes and anchored to room centroids, ensuring unambiguous spatial and topological containment. This removes errors caused by boundary conditions, wall thickness or geometric inaccuracies. All lights, radiators and sensors are successfully assigned to a room node, with no unconnected devices, confirming the effectiveness of the centroid-based strategy under geometric noise and partial room closures.

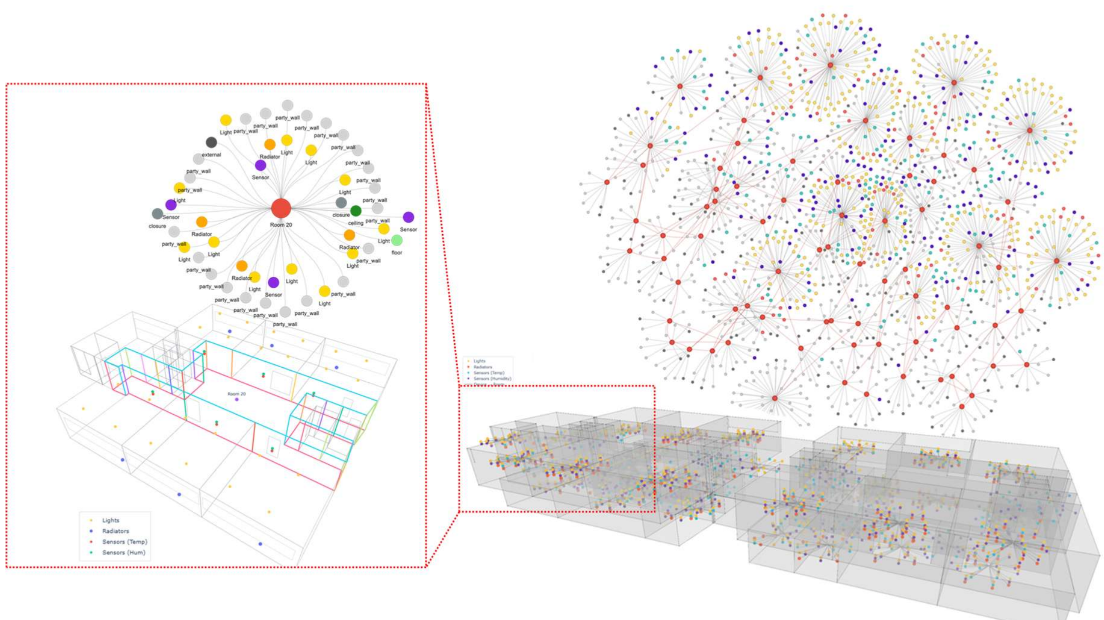


Fig. 90. Room-centric device graphs (left) are embedded in the 3D building geometry and aggregated into a global building graph (right), illustrating consistent device–room associations and scalable spatial connectivity from room to building level.

The resulting graph exhibits a bipartite-like structure, with rooms as core nodes and devices as peripheral nodes, reflecting the hierarchical organization of building systems in Energy DTs. Room–room connectivity encodes spatial adjacency, enabling zone aggregation, thermal coupling, sensor interpolation, fault propagation and graph-based control or optimization. Graph connectivity is preserved within building wings without introducing spurious cross-wing links, demonstrating correct automatic wing separation. The approach is robust to incomplete or degraded geometry, as graph validity relies on topological coherence rather than geometric precision. Scalability is linear with respect to the number of rooms and devices, supporting large buildings and campus-scale Energy DTs. Overall, the graph provides a lightweight abstraction between geometry, sensing and simulation, enabling IoT integration, room-level analytics and simulation-ready Energy DTs while prioritizing functional correctness over geometric exactness.

5.6 Parser–Transformer–Writer Results for IFC, gbXML, and epJSON

5.6.1 gbXML PTW for energy-model interoperability

The gbXML PTW generates analysis-ready XML models compatible with a wide range of building energy tools. The parser reconstructs spatial hierarchy, surface topology, openings, material assignments and georeferencing directly from the graph-based DT (Fig. 91).

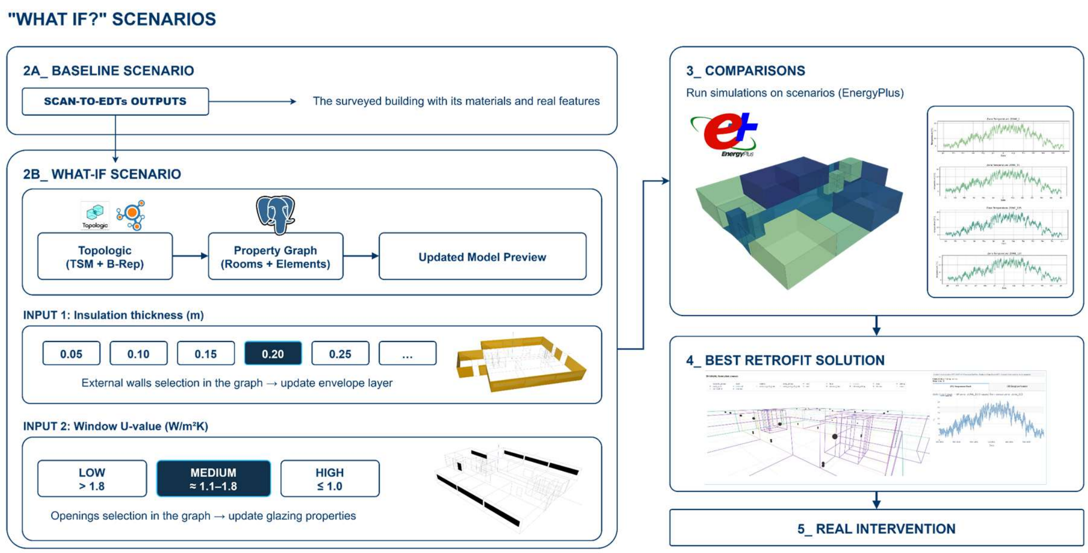


Fig. 91. From the baseline Scan-to-EDT model, graph-driven modifications of envelope properties to generate alternative scenarios, which are simulated and compared to identify the optimal retrofit solution and support real intervention decisions.

The transformer then maps graph entities to gbXML elements in a one-to-one, topology-preserving manner: graph volumes are exported as `<Space>` and `<Zone>` elements; faces and their adjacency relations are converted into `<Surface>` elements with explicit `surfaceType, AdjacentSpaceId` and exposure attributes; openings are mapped to `<Opening>` elements; and materials and layered assemblies are serialized as `<Construction>` and `<Layer>` definitions. Geometry is exported as planar polygonal surfaces with explicit vertex coordinates, while boundary conditions and adjacency are derived directly from the graph topology. The writer produces a schema-valid gbXML document with stable identifiers inherited from the DT, ensuring interoperability and downstream validation. Both PTW pipelines are fully deterministic: identical graph states always generate identical gbXML or epJSON outputs. The modular design supports incremental exports, transactional updates (e.g., retrofit scenarios) and rapid regeneration of simulation inputs without re-parsing geometry. Retrofit operation, such as insulation layer insertion, layer replacement or assignment of target U-values, are applied at the graph level and automatically propagated through the PTW, producing scenario-specific epJSON files ready for execution in EnergyPlus. This enables reproducible baseline-versus-retrofit comparisons and establishes the graph-based DT as the single source of truth for simulation-driven decision support.

To further illustrate the practical utility of the generated models, the energy simulation results obtained from the exported epJSON files and executed in EnergyPlus are summarized below. For the primary validation case study, Santa Chiara Heritage Building, and assuming thermostat setpoints consistent with national regulations (20 °C for heating and 26 °C for cooling), the framework produced a simulation-ready model with an estimated annual heating demand of 0.04 kWh/m² and an annual cooling demand of 0.90 kWh/m². The CFD-based thermal analysis presented in Section 4.8.2 further confirmed spatial temperature distributions within 1–2 °C of the sensor measurements, supporting both the geometric fidelity of the model and the correctness of the thermal boundary conditions assigned through the graph-based material attribution module. Taken together, these results indicate that the topological and geometric quality achieved by the Scan-to-EDT pipeline can be translated into simulation-ready models capable of producing coherent thermal-performance outputs, thereby satisfying the simulation-grade accuracy requirements identified in Chapter 3.

The limited extent of the energy simulation assessment for the remaining datasets is mainly due to the scarce availability of monitoring data in public buildings, as well as privacy-related restrictions affecting private residential case studies.

6

This chapter synthesizes the key outcomes of the thesis by positioning the proposed Scan-to-EDT framework within the broader landscape of Energy Digital Twins in the AEC domain. It first summarizes the main advancements introduced with respect to current practice, highlighting how the pipeline consolidates fragmented processes into a coherent, automation-oriented workflow that remains controllable and reproducible. It then discusses the principal limitations observed across the pipeline—from semantic detection and BIM-oriented reconstruction accuracy to parameter sensitivity in topological enforcement and the constrained availability of indoor IoT data—clarifying how these factors affect scope and performance. Finally, the chapter formalizes the thesis' main original contributions, including the multi-level EDT architecture, the graph-centric internal representation, the hybrid deep learning–deterministic strategy, the object-graph reconstruction approach, and the PTW interoperability layer enabling standards-compliant exports and simulation-ready models. Together, these elements define the methodological novelty and practical impact of the proposed approach.

Chapter 6

6. Advancements, Limitations & Main contributions

The research presented in this PhD thesis advances the state of the art in Energy Digital Twins (EDTs) within the Architecture, Engineering, and Construction (AEC) domain by proposing a unified Scan-to-EDT framework that bridges unstructured survey data and simulation-ready energy models. The work consolidates and extends existing methodologies while introducing original architectural and methodological contributions that enable robust, controllable and reproducible Energy Digital Twin generation.

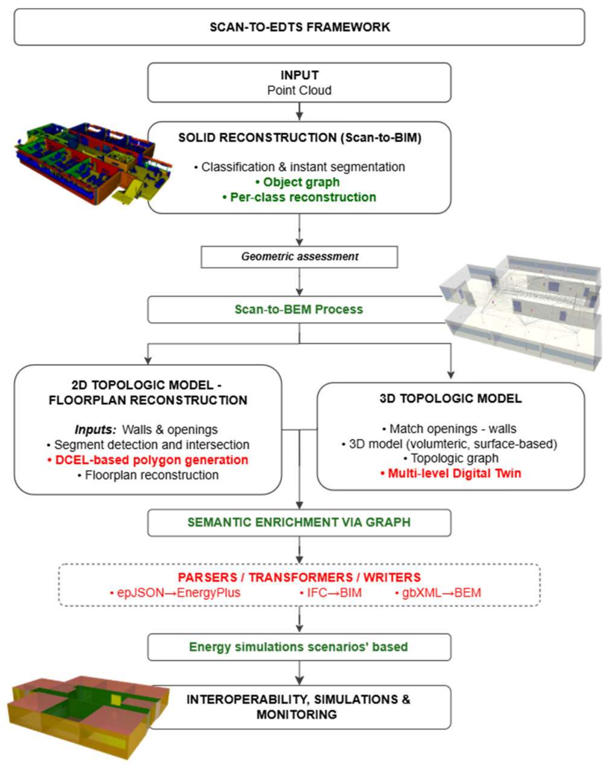


Fig. 92. Scan-to-EDTs schematic framework highlighting advancements (green) and main contributions (red).

In the following sections we critically present:

- **Advancements (Section 6.1)**: The proposed framework introduces methodological advancements that improve automation, reproducibility, and interoperability with respect to existing Scan-to-BIM workflows, topology reconstruction algorithms, and graph-based approaches. These enhancements are integrated throughout the workflow and highlighted in green in the framework diagrams (Fig. 91), supporting more consistent, scalable and transferable processes.
- **Limitations (Section 6.2)**: Constraints and weaknesses that restrict effectiveness, scope, or applicability of the current implementation.
- **Main Contribution (Section 6.3)**: Original innovations that define the core novelty of this work. These are highlighted in red throughout the framework diagrams (Fig. 92).

6.1 Advancements

In the Scan-to-EDTs framework, advancements may be distinguished between those occurring at the methodological level of the main conceptual framework (Section 6.1.1) and those related to specific methods or algorithms (Fig. 93).

6.1.1 Advancements at methodological level

At a methodological level, the proposed framework introduces several advancements over current practice, highlighted in green in Figure 93.

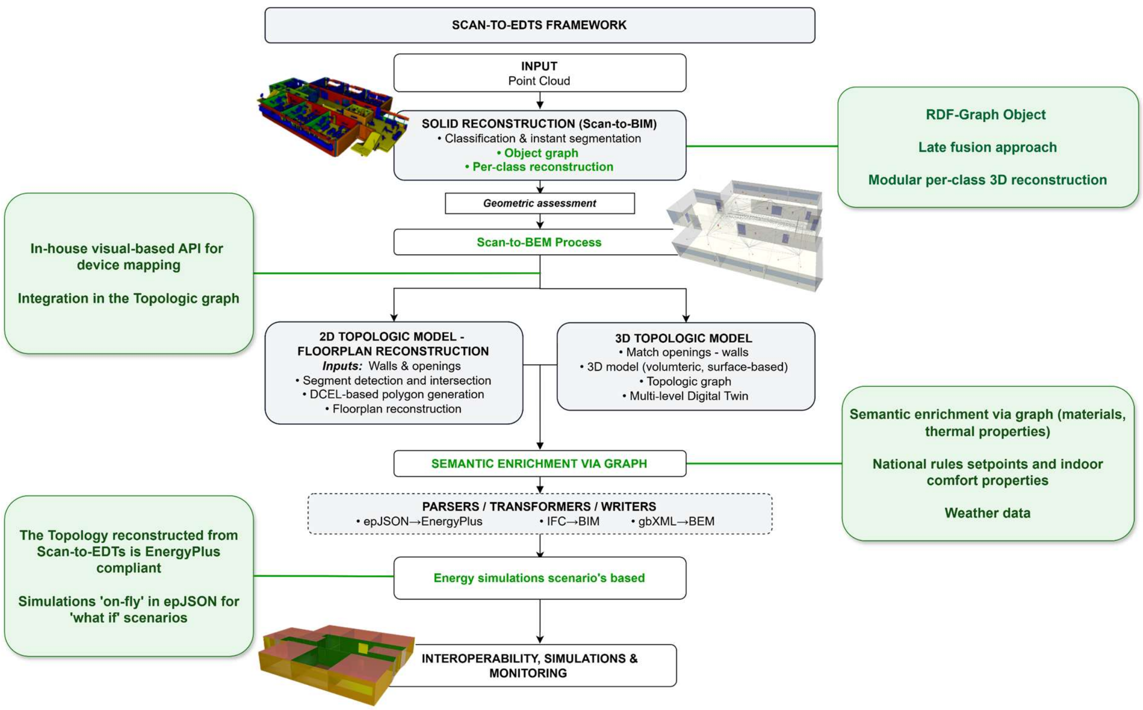


Fig. 93. Overview of the proposed Scan-to-EDT framework, highlighting specific level advancements (in green).

First, previously fragmented workflows are unified into a single end-to-end pipeline encompassing object-graph construction, per-class reconstruction, graph-based semantic enrichment, and simulation-oriented export.

This coherent Scan-to-EDT chain directly links survey data to simulation-ready models, reducing redundancy, improving traceability, and enabling end-to-end automation.

Second, enhanced automation with controlled model generation is achieved through object-graph reasoning, per-class reconstruction and Scan-to-BEM processing, which combine deterministic rule-based modeling with explicit topological enforcement. This approach minimizes manual intervention while ensuring geometric and topological consistency across 2D and 3D representations.

Third, interoperability is strengthened through parser–transformer–writer (PTW) modules that treat IFC, gbXML and epJSON strictly as export formats rather than internal working states. This design limits information loss avoids schema-driven constraints and ensures consistent and reproducible simulation inputs.

Collectively, these methodological advancements form the core of the proposed Scan-to-EDT framework, improving automation, reproducibility, and interoperability while maintaining engineering-grade control over geometry, topology, and semantics.

6.1.2 Advancements at specific level

At the specific level, advancements include:

(i) the application of a late-fusion approach combining point-cloud and image-based data to improve object detection and instance segmentation.
(ii) The use of the *Geomapi* framework for RDF graph generation from point clouds, enabling per-class reconstruction and module-based processing as reported in the CVPR 2024 Challenge repository[1].
(iii) The development of a lightweight interactive API for point-cloud navigation and manual detection of energy-related devices, supporting spatially aware annotation and room-level assignment.
(iv) The construction of a *NetworkX* and *PostgreSQL*-based property graph to represent building components and their topology, enabling material assignment and the generation of simulation-ready models, which constitutes one of the primary objectives of the work.

6.2 Limitations of the framework

Despite the overall robustness of the proposed Scan-to-EDT framework, several limitations remain. These limitations can be g FME software-based representation grouped into four main typologies:

(i) limitations in object detection and semantic classification (Section 6.2.1),
(ii) limitations in BIM-oriented geometric modelling accuracy (Section 6.2.2),
(iii) limitations related to parameter sensitivity and topological enforcement (Section 6.2.3), and
(iv) limitations in the usage and availability of indoor IoT sensor data (Section 6.2.4).

It is important to note that these limitation typologies reflect both the scope and the intent of this research, in particular:

- object detection and semantic classification (i) were implemented and evaluated as enabling components of the Scan-to-EDT pipeline, but their in-depth optimization was not a primary objective of the thesis.
- Limitations in BIM geometric modelling accuracy (ii) are therefore largely dependent on the quality of upstream semantic detection and propagate from it.
- Limitations related to parameter sensitivity and topological enforcement (iii) correspond to well-known engineering challenges in deterministic geometric modelling, especially when operating across heterogeneous building typologies (i.e., heritage buildings, or very dense wall layout buildings, such as schools) and scales.
- Finally, limitations in the usage and availability of indoor IoT sensor data (iv) are mainly due to the emerging nature of Energy DTs, the lack of long-term and complete datasets combining geometry and operational data, and ongoing constraints related to data availability and privacy.

6.2.1 Object Detection and Semantic Classification

Quantitative evaluation shows heterogeneous performance across object classes. Large planar elements such as floors, ceilings, and walls achieve consistently high accuracy (mIoU > 87%, mAcc > 89%), confirming the robustness of the semantic segmentation pipeline for dominant building components. In contrast, smaller, geometrically complex, and often occluded elements remain challenging. Columns exhibit the lowest performance (mIoU ≈ 44%, mAcc ≈ 71%), reflecting systematic difficulty caused by limited surface extent, occlusions, and geometric similarity to adjacent elements. Doors and windows show moderate segmentation accuracy (mIoU 51–59%), mainly due to boundary ambiguity, partial visibility, and architectural variability.

Figure 94 shows an instance-segmentation error on structural columns: a single physical column is mistakenly split into two point-cloud clusters with different object IDs, leading the pipeline to reconstruct two

[1] https://github.com/Saiga1105/Scan-to-BIM-CVPR-2024

overlapping columns in the same location. The reconstruction errors (Fig. 93) highlight how small segmentation inaccuracies can propagate into geometric inconsistencies.

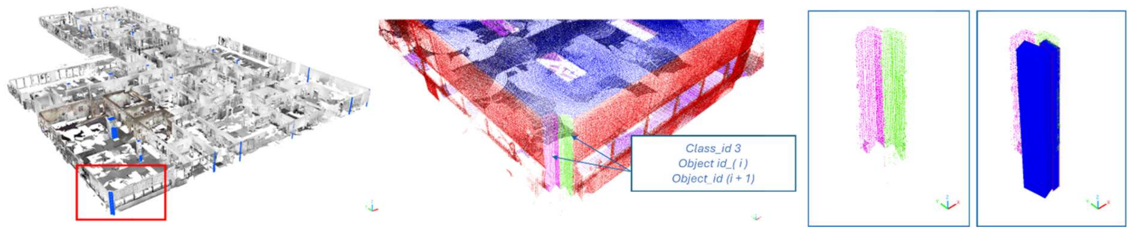


Fig. 94. Column inaccurately mapped and reconstructed within the model due to detection inaccuracies.

As shown in Table 15, segmentation performance could improve with larger and more balanced training datasets, particularly for secondary elements, as the adopted PTv3 architecture is expected to benefit from increased data availability. However, semantic segmentation optimization was not the primary focus of this thesis, which instead targets the development of a robust Scan-to-EDT pipeline for simulation-ready Energy Digital Twins. Consequently, these limitations primarily affect non-dominant components and do not compromise the framework's core objectives.

6.2.2 BIM-Oriented Modelling Precision and Accuracy

Limitations in semantic segmentation propagate directly to BIM-oriented geometric reconstruction. Across multiple datasets and building typologies, walls achieve high modelling accuracy (mIoU and F1 typically between 88% and 96%), confirming the reliability of envelope reconstruction. Conversely, doors, windows, and columns exhibit reduced accuracy (mIoU generally between 67% and 77%), highlighting the dependence of geometric modelling on upstream detection quality and the simplifications inherent in parametric BIM representations.

Certain architectural elements, such as terraces, cannot currently be modelled with sufficient accuracy, as their reconstruction relies on the generation of floors from perimetral walls. Variations in acquisition conditions, point density, and occlusion—e.g., indoor NavVis scans versus large-scale campus environments—further affect modelling precision. While overall performance remains stable, fine-grained reconstruction is less robust than envelope-level modelling.

6.2.3 Parameter Sensitivity and Topological Enforcement

An additional limitation concerns the sensitivity of certain deterministic algorithms to scale-dependent parameters. In particular, the iterative wall–wall intersection detection described in Section 4.4.2 requires calibration with respect to building size and geometric scale to operate fully automatically. Inadequate tuning may introduce minor geometric inconsistencies, especially in buildings with atypical dimensions or variable wall thicknesses.

This scale dependency limits full automation of the Scan-to-EDT pipeline, as parameters effective for one building typology may not generalize to others. Although iterative refinement improves robustness over single-pass detection, the process remains partially heuristic.

Resulting unresolved intersections propagate to the DCEL construction (Section 4.4.5), where they may produce topological artifacts such as dangling edges. These are corrected to preserve watertightness and topological consistency, at the cost of minor geometric imprecision, which can occasionally affect room reconstruction, wall accuracy, or opening preservation.

This reflects a deliberate trade-off favoring topological correctness and simulation readiness over strict geometric fidelity in edge cases.

6.2.4 Indoor IoT Sensor Data Availability

Finally, the practical deployment of the framework is constrained by the limited availability of indoor IoT sensor data. In many real-world settings, sensor measurements are not stored long-term or are available only for short monitoring campaigns. The lack of complete, synchronized datasets combining survey data and operational measurements limits comprehensive validation and generalizability.

Privacy and data protection requirements further restrict access to indoor sensor data, particularly in occupied buildings, reducing spatial and temporal resolution.

As a result, while the framework supports sensor integration, its application is currently limited more by data availability and accessibility than by methodological constraints.

6.2.5 Implications and Outlook

These limitations do not compromise the generation of simulation-ready Energy DTs, as energy analysis primarily depends on envelope geometry, zoning, and material properties. However, they may affect applications requiring high-fidelity as-built BIM models. Future work will address these limitations through improved class balancing, multi-scale learning, adaptive scale-aware parameterization, and tighter integration between semantic inference, geometric validation, and indoor sensor data to support more comprehensive DT calibration.

6.3 Main contribution

Beyond methodological advancements, this work introduces several original contributions that define the core novelty of the proposed Scan-to-EDT framework (Figure 95).

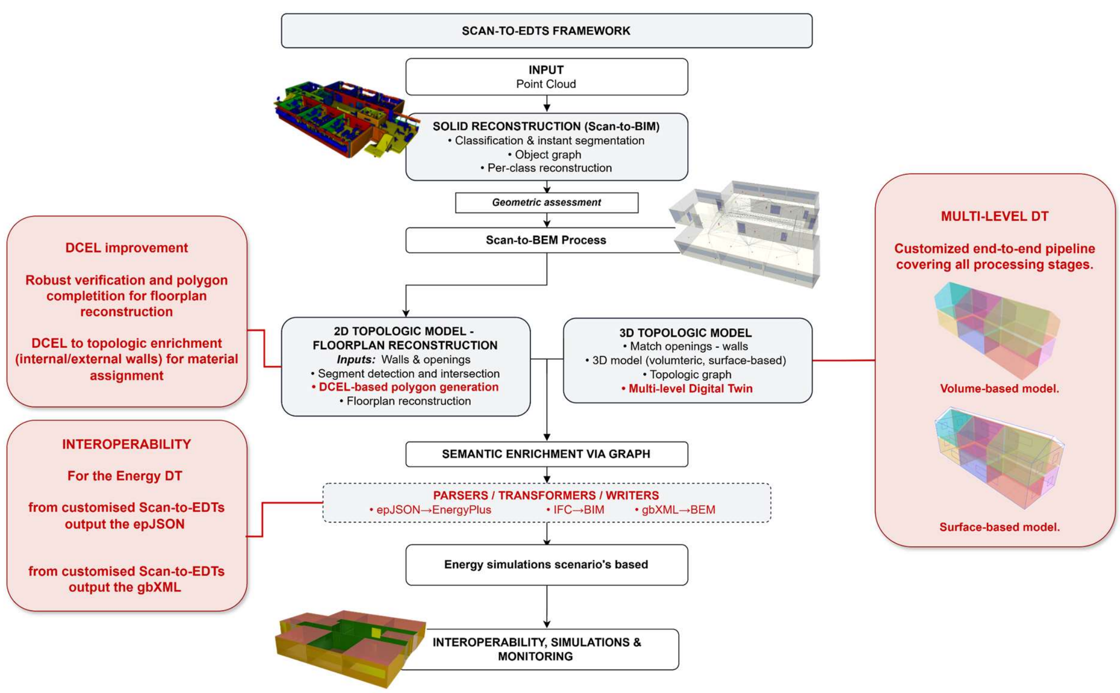


Fig. 95. Overview of the proposed Scan-to-EDT framework, highlighting main contributions (in red).

(i) Multi-level Energy Digital Twin architecture

The research introduces a **multi-level Energy Digital Twin** architecture that unifies volumetric and surface-based representations within a graph-based model. This layered structure enables robust topology construction, supports interoperability across data standards and provides simulation-ready representations with explicit control over geometry, topology and semantics across scales.

(ii) Graph-centric internal representation

A graph-based internal representation is introduced to encode adjacency, containment, connectivity, and watertightness relationships. This representation ensures geometric validity and semantic consistency throughout the pipeline, from raw survey data to simulation-ready Digital Twins.

(iii) Object-graph-based building reconstruction

The work proposes a building reconstruction method (Roman et al., 2024b) based on object-graph representations and per-class reconstruction strategies. This approach supports structured, class-specific geometry generation while maintaining global topological coherence.

(iv) Hybrid deep learning–deterministic workflow

A hybrid workflow is developed in which DL techniques are employed for semantic classification and instance segmentation, while deterministic algorithms enforce geometric reconstruction, topological correctness and model validation. This combination ensures both automation and reliability, addressing the limitations of purely data-driven approaches in engineering contexts.

(v) Parser–Transformer–Writer (PTW) interoperability strategy

The framework introduces a parser–transformer–writer strategy that decouples internal representations from exchange formats. This approach enables controlled, standards-compliant interoperability with IFC, gbXML and epJSON, while preserving semantic richness and minimizing data loss.

(vi) Direct Scan-to-EDT energy simulation workflow

A direct Scan-to-EDT pathway is established for energy simulation, eliminating the need for manual BIM authoring, repeated remodeling, or reliance on proprietary tools. The workflow supports interactive device labeling, automatic semantic enrichment through material and system assignment, and dynamic scenario definition through on-the-fly modification of model parameters.

Collectively, these pipeline-aligned contributions establish a robust, reproducible, and simulation-oriented Digital Twin framework capable of bridging unstructured survey data and operational energy analysis within a single coherent system.

7

This section revisits the research objectives and questions introduced in Chapter 1, assessing how they have been addressed through the development and evaluation of the proposed Scan-to-EDT framework. The overarching objective of this thesis was to formalize, implement, and validate a robust pathway from unstructured survey data to simulation-ready Energy Digital Twins, capable of supporting energy analysis, interoperability, and future operational use.

To this end, five research objectives were defined. The first focused on establishing a clear and operational definition of an Energy Digital Twin, grounded in volumetric, surface-based, and graph-based representations suitable for energy simulation. The second examined the role of BIM within DT workflows, clarifying its function as an enabling but intermediate representation rather than a DT backbone. The third investigated the boundaries of automation and deep learning, identifying where data-driven methods must be complemented by deterministic, topology-aware modeling. The fourth objective defined the functional scope of building DTs, emphasizing simulation, monitoring, and decision support over exhaustive geometric fidelity. Finally, the fifth objective addressed interoperability, proposing graph-driven PTW strategies to ensure reproducible, low-loss exchanges across BIM and energy simulation formats. Together, these objectives guided the design choices of the framework and are reflected in the results presented throughout this thesis.

Chapter 7

7. Conclusions

This research presents a Scan-to–Energy Digital Twin framework (schematically summarized in Fig. 97) that advances the state of the art by enabling the direct transformation of unstructured point clouds into simulation-ready Energy Digital Twins (EDTs) through a scalable and open-source Scan-to-BEM pipeline, eliminating or reducing the need for intermediate manual modeling or BIM authoring stages.

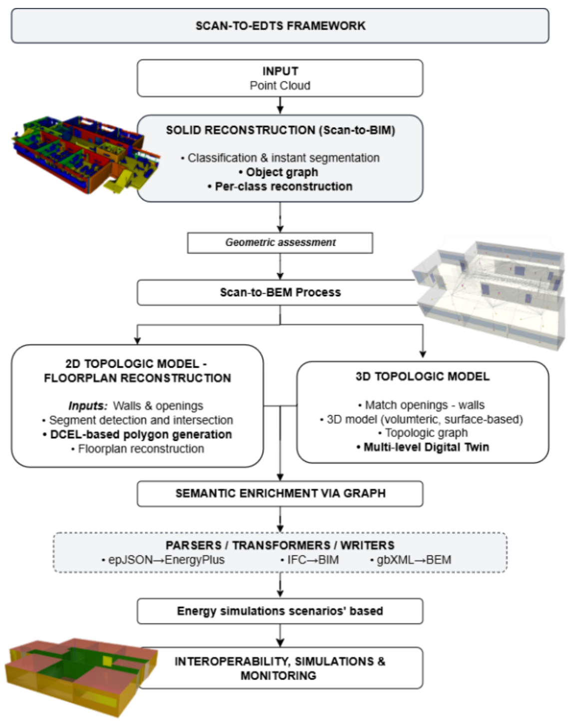


Fig. 96. Schematic overview of the Scan-to-EDT framework.

To address the issues outlined in Chapter 3, the framework is designed around four key principles:

- **Open-source workflow.**
  All core components rely on widely used open-source libraries (e.g., *Point Transformer* (Wu et al., 2024), *Geomapi* (Bassier et al., 2024), *TopologicPy* (https://topologic.app/topologicpy_doc/), *Shapely* (Gillies et al., 2025), *IfcOpenShell* (https://docs.ifcopenshell.org/index.html), and *EnergyPlus* (Naeem et al., 2025)). This avoids software licensing costs, although platform constraints remain (local execution, no cloud-native deployment and a focus on historical rather than real-time data).
- **Customized-format file and dataframe-centred processing.**
  File formats are treated as outputs rather than primary working representations. Internally, processing is centred on tabular data structures (pandas DataFrames) and simple text-based formats (.txt, .csv), while IFC, gbXML and epJSON are generated only at the export stage. This reduces coupling to any single authoring tool or schema.
- **Minimal data loss.**

A single, unified data model underpins the entire pipeline, from scans to simulation-ready models. Parser–transformer–writer (PTW) modules convert between the internal representation and target formats, limiting information loss and avoiding repeated manual re-modelling. Quantitative evidence of reduced data loss is presented in the Results chapter.

- **Automation and reduced manual effort.**
  Most operations, such as geometry reconstruction, topology generation, semantic assignment and export, are automated within the pipeline. Human intervention is mainly confined to configuration and quality control, which reduces repetitive work and the risk of transcription errors. Whenever input data change, the model can be regenerated and re-exported through the PTW modules with minimal manual effort.

In this way, the Scan-to-EDTS framework provides a structured method that can support multiple DT-related operations on the same underlying model, such as (i) monitoring and analyzing historical performance data (and, in future work, real-time streams), (ii) generating and updating simulation-ready geometries, and (iii) progressively enriching the DT through graph-based linkage to additional datasets (Fig. 97).

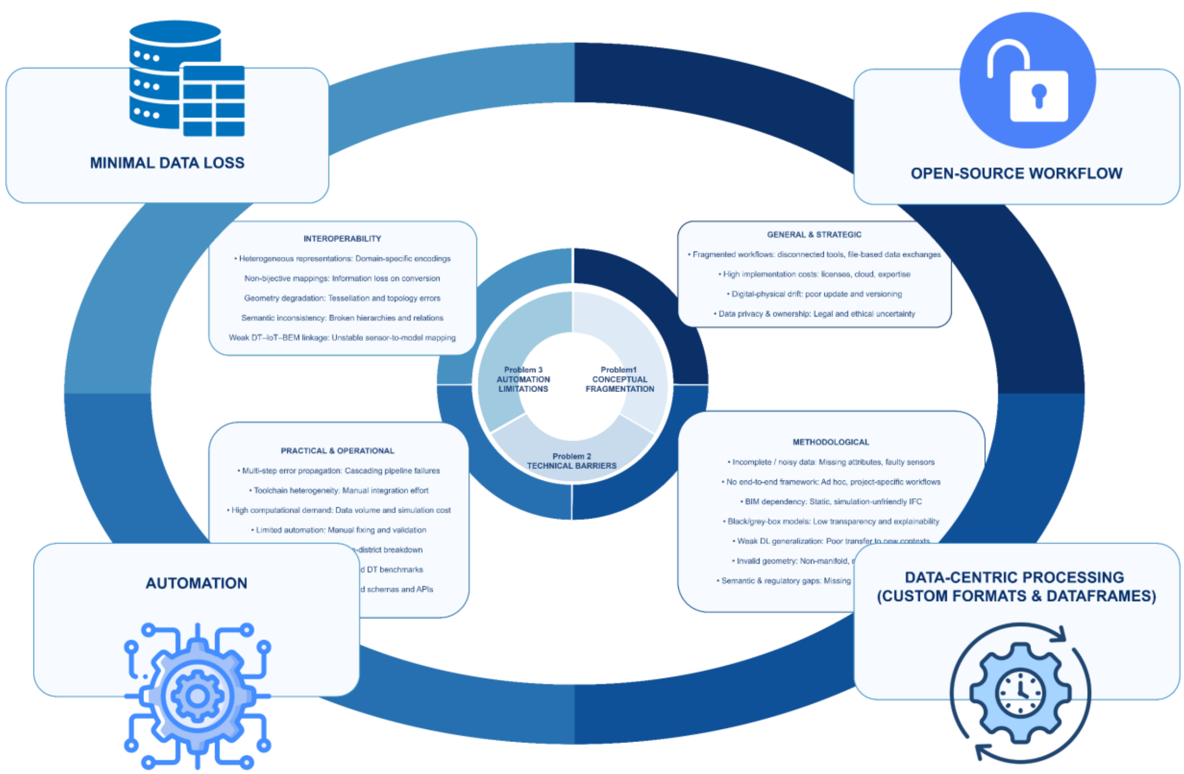


Fig. 97. Conceptual diagram summarizing the main challenges identified in Chapter 1 and the corresponding issues addressed in Chapter 3 (Fig. 21), highlighting key solution themes including automation, data-centric processing, minimal data loss and open-source workflow.

7.1 Overview of the work

The proposed modular framework builds upon (i) robust Scan-to-BIM workflows by integrating state-of-the-art DL networks for object detection, classification and instance segmentation, while extending them with innovative components such as (ii) the in-house enhanced DCEL-based topological modeling and (iii) topologic modelling, with a (iv) parser–transformer–writer (PTW) modules to mitigate data loss and interoperability challenges. By leveraging customized intermediate data formats and graph-based data structures for internal processing, the framework ensures consistent management of geometric, semantic and topological information while enabling flexible, standards-compliant export to multiple target representations.

The framework introduces a multi-level Energy Digital Twin architecture that coherently integrates geometric reconstruction, topological consistency, and semantic enrichment across hierarchical levels of abstraction. The Digital Twin encodes room volumes, semantic surfaces, and geometric elements as mutually consistent entities, ensuring watertight geometry and topological validity at each level.
A property-graph representation serves as the unifying structural mechanism to organize, manage, analyze, edit, and relate these elements. This multi-level architecture supports both solid and boundary representation (B-Rep) models, enabling energy simulation, semantic querying, graph-based navigation, real-time device integration, and standards-compliant exchange via gbXML, CityGML (Energy ADE), and optionally IFC, while minimizing data loss across energy and BIM workflows.

The principal contribution of this research is an end-to-end, interoperable pipeline that transforms raw point cloud data into energy-aware Digital Twins through a unified multi-level, multi-layer modeling approach. By integrating point cloud semantics, topologically consistent building representations, and Building Energy Model (BEM) interoperability within a graph-based internal structure, the framework establishes seamless linkage between geometric, semantic, physical, and operational layers. This architecture enables integration of IoT sensor data—both historical and real-time—while facilitating coherent analysis, simulation, and feedback generation across scales, thereby ensuring consistency, extensibility, and interoperability throughout the DT lifecycle.

7.2 Objectives and research questions

This section aims to synthesize the contributions of this work by revisiting the research questions and the main issues identified in Chapter 1, and by examining the methodological, technical, and interoperability-related challenges discussed in Chapter 3, including the limitations of existing tools. Building on these elements, the section provides explicit responses to the research questions while recalling the objectives that this study sought to achieve. Through this synthesis, the extent to which the proposed approach addresses the initial problem statement is assessed, and the remaining limitations and perspectives for future work are highlighted.

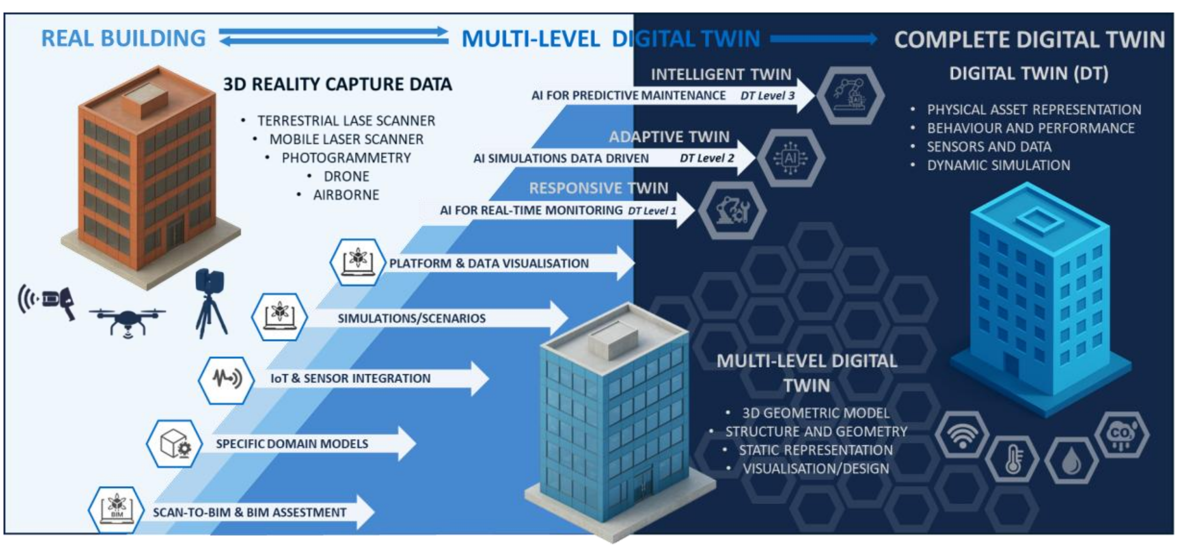


Fig. 98. Core concepts and geometric data structures of the multi-level Digital Twin.

Objective I: Define Energy Digital Twins

RQ1: *What essential components, data flows, and modeling methods formalize an operational definition of an Energy Digital Twin (EDT) and consistently distinguish it from static BIM or 3D models?*

Based on the state of the art and on the practical barriers and limitations identified both prior to and during the development of the Scan-to-EDTs framework, a set of clear design requirements was defined to guide the generation of simulation-ready EDTs. Starting from the target application domain, namely building energy simulation in EnergyPlus and other BEM tools, as well as CFD-based analyses, the framework was designed to satisfy the following core objectives:

- **A volumetric representation** explicitly modeling enclosed thermal zones and volumes V, suitable for physically based energy simulations and heat balance calculations.
- **A surface-based boundary representation (B Rep)** that is watertight, manifold, and topologically consistent, enabling reliable surface discretization for energy and airflow simulations.
- **A unified model capable of supporting both real-time and historical analyses**, allowing the integration of sensor data, simulation outputs, and temporal reasoning within the same Digital Twin structure.

The definition of a multi-level DT (Fig. 98), integrating volume-based representations, surface-based boundary models and a property type relational graph, provides a unified solution capable of efficiently responding to the requirements of energy simulation, analysis, and reasoning. By combining enclosed thermal volumes, watertight and manifold surfaces, and an explicit graph encoding adjacency, containment and functional relationships, this multi-level structure enables robust energy modeling, scalable simulation workflows, and seamless integration of geometric, semantic, and temporal information. This architecture defines the core vision of the proposed EDT, bridging raw spatial data and simulation-ready energy models within a single coherent framework.

Objective II: Clarify BIM's Role in DT Development

RQ2: *Under which conditions is BIM necessary for DT creation, and when can alternative models (mesh-based, graph-based, city-scale) serve as foundations?*

RQ3: *How do BIM-derived representations compare with alternatives in geometric accuracy, semantic richness, and suitability for integrating real-time data streams in DT environments?*

As discussed in Chapters 1 and 3, BIMs are inherently static representations.
While they are highly effective for design, construction, and facility management, they are not conceived to function as the core backbone of a DT. In practice, BIM models must be exported to external platforms to support simulation or dynamic analysis, a process that typically involves format transformations and partial information loss, after which the resulting representations no longer fully belong to the BIM domain nor preserve its original semantic richness.

In contrast, other modeling paradigms, particularly topological and graph-based models, are intrinsically more flexible and better suited to support dynamic and evolving DT applications, where relationships, states and behaviors must be continuously updated and queried.
Nevertheless, Scan-to-BIM and automated BIM-oriented workflows for deriving solid and parametric building models from point clouds remain essential. They provide an efficient and reliable means of generating accurate three-dimensional building representations, which can serve as a geometric substrate for the extraction of higher-level, topology-driven structured models. In this sense, Scan-to-BIM processes play a crucial enabling role, bridging raw survey data and more expressive, dynamic DT representations while supporting downstream energy and simulation workflows.

Therefore, even though BIM-based models are not ideally suited to serve as the backbone of a Digital Twin, the geometric representations derived from decades of BIM and CAD research remain a fundamental asset. These geometries provide a robust and well-established foundation from which more advanced, topology-driven, and semantically enriched models can be derived and further elaborated. From this perspective, BIM geometry should be understood not as the final DT representation, but as a crucial intermediate component, which in the present work is explicitly used to enable the construction of more expressive, dynamic, and simulation-ready Digital Twin models.

Objective III: Establish Automation and AI Boundaries

RQ4: *What can DL reliably extract from point clouds and images, and where does it fail in terms of topology, accuracy, and semantic consistency for engineering applications?*

RQ5: *How can DL outputs be integrated with deterministic methods to produce simulation-ready, DT-ready models with guaranteed geometric validity?*

From the results of this work, DL can reliably extract semantic labels and instance level cues from point clouds and images, including walls, floors, columns, openings and their approximate spatial extents and coarse geometry, even in large and heterogeneous scenes.
However, DL methods do not yet completely guarantee topological correctness or geometric validity. Typical failure modes include fragmented instances, missing or duplicated elements, inaccurate boundaries and inconsistencies in adjacency, connectivity and enclosure.
As a result, DL outputs alone remain insufficient for engineering-grade applications, such as energy simulation, where watertightness, manifold geometry and robust explicit topology are required. In this sense, DL performs well as a perceptual and semantic extractor, but not as a standalone modeling solution, even though recent works (Mayer et al., 2025) are increasingly closing this gap.

Objective IV: Define DT Functional Scope

RQ6: *What core functions should building DTs fulfill, and which abstractions of geometry, semantics, and system behavior balance model complexity with practical usability?*

RQ7: *What mechanisms ensure DTs remain accurate, evolving proxies of physical assets rather than static snapshots?*

Closely linked to RQ1, this work emphasizes that physical reality is inherently complex and cannot be fully replicated digitally; therefore, the functional objectives of a building Energy DT must be explicitly defined and constrained by its intended use, namely simulation, monitoring, querying, and decision support.
The required abstractions of geometry, semantics, and system behavior follow directly from these objectives.
Consequently, exchange file formats are treated as outputs rather than core representations, ensuring that robust, standards compliant internal models are preserved for reliable interpretation, reasoning, and energy simulation.
The use of these models within specific simulation workflows, software, or tools is then derived accordingly, as demonstrated in this framework through parser transformer writer modules, providing full control over engineering usability and practical applicability.

Objective V: Ensure Robust Interoperability

RQ8: Which modeling and exchange strategies minimize data loss and preserve geometric/semantic integrity across BIM, simulation, and DT workflows?

RQ9: How can automated validation, correction, and synchronization prevent repeated remodeling and maintain reliable, reproducible dataflows?

In the proposed framework, data loss is minimized by adopting a unified internal representation based on topological graphs, in which geometry, semantics, and relationships are explicitly modeled and decoupled from

exchange formats. BIM is used as a controlled geometric and semantic source, where attributes such as object classes, thicknesses, positions, and orientations are explicitly defined and validated.
Topological structures then encode adjacency, containment, connectivity, and watertightness, ensuring consistency across representations. Exchange formats such as IFC, gbXML, and epJSON are treated strictly as outputs, generated through dedicated parser transformer writer modules, thereby preserving geometric and semantic integrity from survey and classification to the final Energy DT and simulation models.

The framework ensures reliability and reproducibility through stepwise validation and deterministic control at each stage of the pipeline. Automated checks enforce geometric validity, topological consistency, and semantic completeness before export, while the graph-based core acts as a single source of truth for synchronization across BIM, simulation and DT representations. This approach eliminates repeated manual remodeling, enables consistent regeneration of outputs when inputs change, and guarantees traceable, reproducible dataflows suitable for engineering-grade simulation and analysis.

7.3 Future applications and research directions

Thanks to its flexibility, adaptability, and scalability, the proposed method can be readily extended to (i) different spatial scales and (ii) other structural and infrastructure typologies. By leveraging the underlying graph structure, the approach goes beyond purely geometric analysis and can be applied to energy-related tasks, as well as to structural assessment or monitoring applications.
In this direction, the proposed approach has been tested on bridge infrastructure (Gaspari et al., 2025), where the scan-to-mesh pipeline was applied to a bridge, producing a geometric model subsequently transferred into an IFC representation and visualized using the Xeokit (https://github.com/xeokit/xeokit-sdk) framework. This model can be further refined in future work to support structural assessment, monitoring, and infrastructure management tasks.

A promising research direction lies in applying this method to city model enrichment, a domain in which large-scale urban representations are becoming increasingly important. The graph-based formulation allows the approach to dynamically adapt to the available data. For instance, when only LiDAR point clouds are available, buildings can be reconstructed primarily from façades and roofs, typically without explicitly modeling openings, which remain difficult to detect and reliably classify at this scale.
This flexibility enables the generation of consistent multi-level urban models and multi-level Digital Twins suitable for downstream applications, including the derivation of gbXML representations, the generation of IFC building shells, as explored in recent studies (van der Vaart, 2025), and the production of epJSON *VertexArray* models for energy simulation workflows.
The method can be employed either as a stand-alone workflow, implementing the full pipeline presented in this study, or as a post-processing approach applied to manually or automatically generated BIM models or city-scale generated models. These input models are first decomposed into face-based surface elements, which are subsequently processed to construct multi-level Digital Twin representations.

Figure 99 illustrates a first city-scale application of the proposed method. The graph-based structure was employed to repair the geometries of existing LoD2 models obtained from different institutions and repositories, including CityGML Wiki, TU Delft and Towards Data Science. More specifically, the method was used to repair building topology at both (i) the solid level and (ii) the surface-based level. This repair procedure makes the models suitable for use in open-source energy simulation software, such as EnergyPlus, thereby supporting the development of multi-level digital twin models at the urban scale. This work represents an initial outcome of the research activities envisaged within the present PhD thesis and should be regarded as one tested solution within a broader set of approaches that will be explored in future developments.

From an energy simulation perspective, future work will focus on three directions: (i) deeper integration of thermal simulation and heat-flow analysis to evaluate energy efficiency at both building and component levels; (ii) exploration of predictive energy optimization scenarios using AI-based control strategies (Physics Informed Neural Network, PiNN); and (iii) integration of occupancy behaviour models to improve simulation realism and dynamic responsiveness of the EDT.

Fig. 99. First outputs of city level application: the graph structure here is used for repairing LoD2 building models from a topologic point of view.

Finally, based on the state-of-the-art literature, the core idea of this method appears well suited to serve as a backbone for the creation of Building Digital Twins. As widely recognized and shown in Chapter 2, a single representation is insufficient to capture the complexity required even for a specific Digital Twin task. In this context, the proposed approach embeds and interconnects three complementary representations, namely (i) a volume-based model, (ii) a surface-based model and (iii) the associated graph. Together, these representations provide a robust framework for generating models that can be progressively enriched, support simulation feedback loops and enable continuous monitoring, thereby establishing a backbone for the development of increasingly autonomous Digital Twins.

Bibliography

Allen, B.D., 2021. Digital twins and living models at NASA. In: *Proceedings of the Digital Twin Summit*, Virtual, 3–4 November 2021.

Andriasyan, M., Moyano, J., Nieto-Julián, J.E., Antón, D., 2020. From Point Cloud Data to Building Information Modelling: An Automatic Parametric Workflow for Heritage. *Remote Sensing*, 12(7), 1094.

Armijo, A., Zamora-Sánchez, D., 2024. Integration of Railway Bridge Structural Health Monitoring into the Internet of Things with a Digital Twin: A Case Study. *Sensors*, 24, 2115.

Arroyo Ohori, K., Diakité, A., Krijnen, T., Ledoux, H., Stoter, J., 2018. Processing BIM and GIS Models in Practice: Experiences and Recommendations from a GeoBIM Project in The Netherlands. *ISPRS Int.Journal of Geo-Information*, 7(8), 311.

Arroyo Ohori, K., Ledoux, H., Biljecki, F., Stoter, J., 2015. Modeling a 3D city model and its levels of detail as a true 4D model. *ISPRS International Journal of Geo-Information*, 4(3), 1055–1075.

Arsecularatne, B.P., Rodrigo, N., Chang, R., 2024. Review of reducing energy consumption and carbon emissions through digital twin in built environment. *Journal of Building Engineering*, 98, 111150.

Asare, K.A.B., Liu, R., Anumba, C.J., Issa, R.R.A., 2024. Real-World Prototyping and Evaluation of Digital Twins for Predictive Facility Maintenance. *Journal of Building Engineering*, 97, 110890.

Attaran, S., Attaran, M., Celik, B.G., 2024. Digital Twins and Industrial Internet of Things: Uncovering Operational Intelligence in Industry 4.0. *Decision Analytics Journal*, 10, 100398.

Badrinarayanan, V., Kendall, A., Cipolla, R., 2016. SegNet: A deep convolutional encoder–decoder architecture for image segmentation. *arXiv preprint*, arXiv:1511.00561.

Bassier, M., Vergauwen, M., 2020. Unsupervised reconstruction of Building Information Modeling wall objects from point cloud data. *Automation in Construction* 120, 103338.

Bassier, M., Vermandere, J., De Geyter, S., De Winter, H., 2024. GEOMAPI: Processing close-range sensing data of construction scenes with semantic web technologies. *Automation in Construction* 164, 105454.

Belsky, M., Sacks, R., Brilakis, I., 2016. Semantic enrichment for building information modeling. *Computer-Aided Civil and Infrastructure Engineering*, 31(4), 261–274.

Bienvenido-Huertas, D., Nieto-Julián, J.E., Moyano, J.J., Macías-Bernal, J.M., Castro, J., 2019. Implementing artificial intelligence in H-BIM using the J48 algorithm to manage historic buildings. *International Journal of Architectural Heritage*, 14(8), 1148–1160.

Blaha, M., Vogel, C., Richard, A., Wegner, J. D., Pock, T., Schindler, K., 2016. Large-scale semantic 3D reconstruction: An adaptive multi-resolution model for multi-class volumetric labeling. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 3176–3184.

Boguslawski, P., Gold, C., 2016. The dual half-edge—A topological primal/dual data structure and construction operators for modelling and manipulating cell complexes. *ISPRS International Journal of Geo-Information*, 5(1), 19.

Boguslawski, P., Gold, C.M., Ledoux, H., 2011. Modelling and analysing 3D buildings with a primal/dual data structure. *ISPRS Journal of Photogrammetry and Remote Sensing*, 66(2), 188–197.

Buruzs, A., Šipetić, M., Blank-Landeshammer, B., Zucker, G., 2022. IFC BIM Model Enrichment with Space Function Information Using Graph Neural Networks. *Energies*, 15, 2937.

Chen, Z., Shi, Y., Nan, L., Xiong, Z., Zhu, X.X., 2024. PolyGNN: Polyhedron-based graph neural network for 3D building reconstruction from point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing*, 218, 693–706.

Chen, J., Deng, R., Furukawa, Y., 2024. PolyDiffuse: Polygonal shape reconstruction via guided set diffusion models. *Advances in Neural Information Processing Systems*, 36.

Chen, J., Liu, C., Wu, J., Furukawa, Y., 2019. Floor-SP: Inverse CAD for floorplans by sequential room-wise shortest path. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2661–2670.

Chen, J., Qian, Y., Furukawa, Y., 2022. Heat: Holistic edge attention transformer for structured reconstruction. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 3866–3875.

Chen, Z., Ledoux, H., Khademi, S. Nan, L., 2022a. Reconstructing compact building models from point clouds using deep implicit fields. *ISPRS Journal of Photogrammetry and Remote Sensing, 194*, pp.58-73.

Cheng, B., Misra, I., Schwing, A.G., Kirillov, A., Girdhar, R., 2022. Masked-attention Mask Transformer for universal image segmentation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR 2022)*, 1280–1289.

Choy, C., Gwak, J., Savarese, S., 2019. 4D spatio-temporal ConvNets: Minkowski convolutional neural networks. *arXiv preprint*, arXiv:1904.08755.

Cordts, M., Omran, M., Ramos, S., Rehfeld, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., 2016. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3213-3223).

Cummins, L., Sommers, A., Ramezani, S.B., Mittal, S., Jabour, J., Seale, M., Rahimi, S., 2024. Explainable Predictive Maintenance: A Survey of Current Methods, Challenges and Opportunities. *IEEE Access*, 12, 57574–57602.

Dai, A., Nießner, M., 2019. 3D-SIS: 3D semantic instance segmentation of RGB-D scans. *arXiv preprint*, arXiv:1812.07003.

Dai, K., Jiang, Z., Xie, T., Wang, K., Liu, D., Fan, Z., Li, R., Zhao, L., Omar, M., 2025. SOFW: A synergistic optimization framework for indoor 3D object detection. *IEEE Transactions on Multimedia*, 27, 637–651.

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

de Berg, M., Cheong, O., van Kreveld, M., Overmars, M., 2008. *Computational Geometry: Algorithms and Applications*, 3rd ed. Springer, Berlin–Heidelberg.

El Idrissi, B., Baïna, S., Mamouny, A., Elmaallam, M., 2022. RDF/OWL storage and management in relational database management systems: A comparative study. *Journal of King Saud University – Computer and Information Sciences*, 34(9), 7604–7620.

El-Mekawy, M., Östman, A., Hijazi, I., 2012. A unified building model for 3D urban GIS. *ISPRS International Journal of Geo-Information*, 1(2), 120–145.

Erişen, S., Mehranfar, M., Borrmann, A., 2025. Single Image to Semantic BIM: Domain-Adapted 3D Reconstruction and Annotations via Multi-Task Deep Learning. *Remote Sensing*, 17(16), 2910.

Ester, M., Kriegel, H.-P., Sander, J., Xu, X., 1996. A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining (KDD'96)*, Portland, OR, USA, pp. 226–231.

FME Software. Copyright (c) Safe Software Inc. Available online: www.safe.com (accessed on 28 March 2025).

Fol, C.R., Shi, N., Overney, N., Murtiyoso, A., Remondino, F., Griess, V.C., 2024. 3D dataset generation using virtual reality for forest biodiversity. *International Journal of Digital Earth*, 17(1), 2422984.

Gaspari, F., Fascia, R., Barbieri, F., Roman, O., Carrion, D., Pinto, L., 2025. A 3D WebGIS Open-Source Prototype for Bridge Inspection Data Management. *Geomatics*, *5*(4), 68.

Gillies, S., van der Wel, C., Van den Bossche, J., Taves, M.W., Arnott, J., Ward, B.C., et al., 2025. *Shapely* (Version 2.1.1) [software].

Gómez-Gil, M., Espinosa-Fernández, A., López-Mesa, B., 2022. Contribution of New Digital Technologies to the Digital Building Logbook. *Buildings*, 12, 2129.

Graham, B., Engelcke, M., van der Maaten, L., 2017. 3D semantic segmentation with submanifold sparse convolutional networks. *arXiv preprint*, arXiv:1711.10275.

Grieves, M., 2016. Origins of the Digital Twin Concept. doi:10.13140/RG.2.2.26367.61609.

Hemmer, M., 2024. Algebraic Foundations. In: *CGAL User and Reference Manual*, CGAL Editorial Board, 5.6.1 ed.

Hou, J., Dai, A. and Nießner, M., 2019. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 4421-4430.

Hu, W., Fang, J., Zhang, T., Liu, Z., Tan, J., 2023. A new quantitative digital twin maturity model for high-end equipment. *Journal of Manufacturing Systems*, 66, 248–259.

Hu, X., Olgun, G., Assaad, R.H., 2024. An Intelligent BIM-Enabled Digital Twin Framework for Real-Time Structural Health Monitoring Using Wireless IoT Sensing, Digital Signal Processing, and Structural Analysis. *Expert Systems with Applications*, 252, 124204.

Isikdag, U., Zlatanova, S., 2009. Towards defining a framework for automatic generation of 3D city models from BIM. In: Lee, J., Zlatanova, S. (Eds.), *3D Geo-Information Sciences*. Springer, Berlin, Heidelberg, 79–96.

Ji, A., Chew, A.W.Z., Xue, X., Zhang, L., 2022. An Encoder-Decoder Deep Learning Method for Multi-Class Object Segmentation from 3D Tunnel Point Clouds. *Automation in Construction*, 137, 104187.

Kada, M., McKinley, L., 2009. 3D Building Reconstruction from LIDAR based on a Cell Decomposition Approach. *International Archives of Photogrammetry and Remote Sensing*, 38(3/W4).

Kazhdan, M., Bolitho, M., Hoppe, H., 2006. Poisson surface reconstruction. In: *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, 7(4).

Khanam, R., Hussain, M., 2024. YOLOv11: An overview of the key architectural enhancements. *arXiv preprint*, arXiv:2410.17725.

Kim, J., 2020. Visual Analytics for Operation-Level Construction Monitoring and Documentation: State-of-the-Art Technologies, Research Challenges, and Future Directions. *Frontiers in Built Environment* **6**, 575738.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A.C., Lo, W.-Y., Dollár, P., Girshick, R., 2023. Segment Anything. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Paris, France, 1–6 October 2023, 3992–4003.

Kreuzer, T., Papapetrou, P., Zdravkovic, J., 2024. Artificial Intelligence in Digital Twins—A Systematic Literature Review. *Data & Knowledge Engineering*, 151, 102304.

Landrieu, L., Simonovsky, M., 2018. Large-scale point cloud semantic segmentation with superpoint graphs. *arXiv preprint*, arXiv:1711.09869.

Ledoux, H., Arroyo Ohori, K., Kumar, K., et al., 2019. CityJSON: a compact and easy-to-use encoding of the CityGML data model. *Open Geospatial Data, Software and Standards*, 4, 4.

Ledoux, H., Meijers, M., 2011. Topologically consistent 3D city models obtained by extrusion. *International Journal of Geographical Information Science*, 25(4), 557–574.

Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B., 2018. PointCNN: Convolution on X-transformed points. *arXiv preprint*, arXiv:1801.07791.

Liebenberg, M., Jarke, M., 2023. Information Systems Engineering with Digital Shadows: Concept and Use Cases in the Internet of Production. *Information Systems*, 114, 102182.

Lin, T.Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., Dollár, P., Zitnick, C.L., 2014. Microsoft coco: Common objects in context. In *European conference on computer vision* (pp. 740-755). Cham: Springer International Publishing.

Lin, T., Yu, Z., McGinity, M., Gumhold, S., 2024. An immersive labeling method for large point clouds. *Computers & Graphics*, 124, 104101.

Liu, S., Zeng, Z., Ren, T., Li, F., Zhang, H., Yang, J., Li, C., Yang, J., Su, H., Zhu, J., et al., 2023. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. *arXiv preprint*, arXiv:2303.05499.

Liu, Z., Li, M., Ji, W., 2025. Development and Application of a Digital Twin Model for Net Zero Energy Building Operation and Maintenance Utilizing BIM-IoT Integration. *Energy and Buildings*, 328, 115170.

Logg, A., Wells, G., Mardal, K.-A., 2011. *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*. Lecture Notes in Computational Science and Engineering, 84. *Springer*, Berlin, Heidelberg.

Lu, Q., Xie, X., Parlikad, A.K., Schooling, J.M., 2020. Digital twin-enabled anomaly detection for built asset monitoring in operation and maintenance. *Automation in Construction*, 118, 103277.

Maalek, R., Lichti, R., Ruwanpura, J.Y., 2019. Automatic Recognition of Common Structural Elements from Point Clouds for Automated Progress Monitoring and Dimensional Quality Control in Reinforced Concrete Construction. *Remote Sensing*.

Malhotra, A., 2023. DESCity: District energy simulation using CityGML models. Dissertation, RWTH Aachen University, Aachen, Germany. doi:10.18154/RWTH-2023-05656.

ManyCore Research Team, 2025. SpatialLM: Large Language Model for Spatial Understanding. GitHub repository. Available at: https://github.com/manycore-research/SpatialLM (accessed 12 March 2025).

Mayer, K., Vesel, A., Zhao, X., Fischer, M., 2025. SYNBUILD-3D: A large, multi-modal, and semantically rich synthetic dataset of 3D building models at Level of Detail 4. *arXiv preprint*, arXiv:2508.21169.

Meddaoui, A., Hain, M., Hachmoud, A., 2023. The benefits of predictive maintenance in manufacturing excellence: A case study to establish reliable methods for predicting failures. *International Journal of Advanced Manufacturing Technology*, 128, 3685–3690.

Mescheder, L., Oechsle, M., Niemeyer, M., Nowozin, S., Geiger, A., 2019. Occupancy networks: Learning 3D reconstruction in function space. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4460–4470.

Mildenhall, B., Srinivasan, P.P., Tancik, M., Barron, J.T., Ramamoorthi, R., Ng, R., 2020. NeRF: Representing scenes as neural radiance fields for view synthesis. In: *Proceedings of the European Conference on Computer Vision (ECCV 2020)*, Lecture Notes in Computer Science, 12346, Springer, 405–421.

Mousavi, Y., Gharineiat, Z., Karimi, A.A., McDougall, K., Rossi, A., Gonizzi Barsanti, S., 2024. Digital twin technology in built environment: A review of applications, capabilities and challenges. *Smart Cities*, 7(5), 2594–2615.

Naeem, A., Ho, C.O., Kolderup, E., Jain, R.K., Benson, S., de Chalendar, J., 2025. EnergyPlus as a computational engine for commercial building operational digital twins. *Energy and Buildings*, 329, 115257.

Nan, L., Wonka, P., 2017. PolyFit: Polygonal surface reconstruction from point clouds. In: *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, pp. 2353–2361.

Neo4j, Inc., 2024. Neo4j Graph Data Platform. Available at: https://neo4j.com.

OpenCFD Ltd., n.d. OpenFOAM: The Open Source CFD Toolbox. Available at: https://www.openfoam.org.

Ozkan, M., Akin, E., Aslan, Ö., Kosunalp, S., Iliev, T., Stoyanov, I., 2024. A comprehensive survey: Evaluating the efficiency of artificial intelligence and machine learning techniques on cyber security solutions. *IEEE Access*.

Pantoja-Rosero, B.G., Rusnak, A., Kaplan, F., Beyer, K., 2024. Generation of LOD4 models for buildings towards the automated 3D modeling of BIMs and digital twins. *Automation in Construction*, 168, 105822.

Park, J. J., Florence, P., Straub, J., Newcombe, R., Lovegrove, S., 2019. DeepSDF: Learning continuous signed distance functions for shape representation. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 165–174.

Pavirani, F., Gokhale, G., Claessens, B., Develder, C., 2024. Demand Response for Residential Building Heating: Effective Monte Carlo Tree Search Control Based on Physics-Informed Neural Networks. *Energy and Buildings*, 311, 114161.

Perez-Perez, Y., Golparvar-Fard, M., El-Rayes, K., 2021. Scan2BIM-NET: Deep Learning Method for Segmentation of Point Clouds for Scan-to-BIM. *Journal of Construction Engineering and Management*, 147(9), 04021107.

Poudel, R.P.K., Liwicki, S., Cipolla, R., 2019. Fast-SCNN: Fast semantic segmentation network. *arXiv preprint*, arXiv:1902.04502.

Qi, C.R., Su, H., Mo, K., Guibas, L.J., 2017. PointNet: Deep learning on point sets for 3D classification and segmentation. *arXiv preprint*, arXiv:1612.00593.

Qi, C.R., Yi, L., Su, H., Guibas, L.J., 2017. PointNet++: Deep hierarchical feature learning on point sets in a metric space. *arXiv preprint*, arXiv:1706.02413.

Qian, G., Li, Y., Peng, H., et al., 2022. PointNeXt: Revisiting PointNet++ with improved training and scaling strategies. *arXiv preprint*, arXiv:2206.04670.

Ren, D., Li, J., Wu, Z., Guo, J., Wei, M., Guo, Y., 2024. MFFNet: Multimodal Feature Fusion Network for Point Cloud Semantic Segmentation. *The Visual Computer*, 40, 5155–5167.

Robert, D., Raguet, H., Landrieu, L., 2023. Efficient 3D semantic segmentation with superpoint transformer. *arXiv preprint*, arXiv:2306.08045.

Robert, D., Raguet, H., Landrieu, L., 2024. Scalable 3D panoptic segmentation as Superpoint graph clustering. *arXiv preprint*, arXiv:2401.06704.

Roman, O., Mazzacca, G., Farella, E. M., Remondino, F., Bassier, M., and Agugiaro, G., 2024a. Towards Automated BIM and BEM Model Generation using a B-Rep-based Method with Topological Map, *ISPRS Annals Photogramm. Remote Sens. Spatial Inf. Sci., X-4-2024*, 287–294.

Roman, O., Bassier, M., De Geyter, S., De Winter, H., Farella, E. M., Remondino, F., 2024b. BIM Module for Deep Learning-driven parametric IFC reconstruction, *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLVIII-2/W8-2024*, 403–410.

Roman, O., Farella, E.M., Remondino, F., 2025a. Simulation-Driven Thermal Analyses for Cultural Heritage Conservation. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, X-M-2–2025, 301–308.

Roman, O., Farella, E. M., and Remondino, F., 2025b. Simulation-driven Thermal Analyses for Cultural Heritage Conservation, *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., X-M-2-2025*, 301–308.

Roman, O., Bassier, M., Agugiaro, G., Arroyo Ohori, K., Farella, E. M., Remondino, F., 2025c. Scan-to-EDTs: Automated Generation of Energy Digital Twins from 3D Point Clouds. *Buildings*, 15, 4060.

Ronneberger, O., Fischer, P., Brox, T., 2015. U-Net: Convolutional networks for biomedical image segmentation. In: Navab, N., Hornegger, J., Wells, W., Frangi, A. (Eds.), *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, Lecture Notes in Computer Science, 9351. Springer, Cham, 234–241.

Samadzadegan, F., Toosi, A., Dadrass Javan, F., 2024. A critical review on multi-sensor and multi-platform remote sensing data fusion approaches: current status and prospects. *International Journal of Remote Sensing*, 46(3), 1327–1402.

Sharma, A., Kosasih, E., Zhang, J., Brintrup, A., Calinescu, A., 2022. Digital twins: State of the art theory and practice, challenges, and open research questions. *Journal of Industrial Information Integration*, 30, 100383.

Shi, W., Ragunathan, R., Rajkumar, R., 2020. Point-GNN: Graph neural network for 3D object detection in a point cloud. *arXiv preprint*, arXiv:2003.01251.

Shi, X., Wang, S., Zhang, B., Ding, X., Qi, P., Qu, H., Li, N., Wu, J., Yang, H., 2025. Advances in object detection and localization techniques for fruit harvesting robots. *Agronomy*, 15(1), 145.

Skrzypczak, I., Oleniacz, G., Leśniak, A., Zima, K., Mrówczyńska, M., Kazak, J.K., 2022. Scan-to-BIM method in construction: assessment of the 3D buildings model accuracy in terms inventory measurements. *Building Research & Information*, 50(8), 859–880.

Stekovic, S., Rad, M., Fraundorfer, F., Lepetit, V., 2021. MonteFloor: Extending MCTS for reconstructing accurate large scale floor plans. *arXiv preprint*, arXiv:2103.11161.

Strudel, R., Garcia, R., Laptev, I., Schmid, C., 2021. Segmenter: Transformer for semantic segmentation. *arXiv preprint*, arXiv:2105.05633.

Su, J.-W., Tung, K.-Y., Peng, C.-H., Wonka, P., Chu, H.K., 2023. SLIBO-Net: Floorplan reconstruction via slicing box representation with local geometry regularization. *Advances in Neural Information Processing Systems*, 36.

Sun, C., Han, J., Deng, W., Wang, X., Qin, Z., Gould, S., 2025a. 3D-GPT: Procedural 3D modeling with large language models. In: *Proceedings of the International Conference on 3D Vision (3DV)*, Singapore, 1253–1263.

Sun, Q., Fang, C., Liu, S., Sun, Y., Shang, Y., He, Y., 2025b. PolyGraph: A graph-based method for floorplan reconstruction from 3D scans. *IEEE Transactions on Visualization and Computer Graphics*, 31(10), 7350–7362.

Szrek, A., Romańczukiewicz, K., Kujawa, P., Trybała, P., 2024. Comparison of TLS and SLAM technologies for 3D reconstruction of objects with different geometries. *IOP Conference Series: Earth and Environmental Science*, 1295(1), 012012.

Tang, P., Huber, D., Akinci, B., Lipman, R., Lytle, A., 2010. Automatic reconstruction of as-built building information models from laser-scanned point clouds: A review of related techniques. *Automation in Construction*, 19(7), 829–843.

Thomas, H., Qi, C.R., Deschaud, J.-E., Marcotegui, B., Goulette, F., Guibas, L.J., 2019. KPConv: Flexible and deformable convolution for point clouds. *arXiv preprint*, arXiv:1904.08889.

Tuhaise, V.V., Tah, J.H.M., Abanda, F.H., 2023. Technologies for Digital Twin Applications in Construction. *Automation in Construction*, 152, 104931.

van der Vaart, J., Arroyo Ohori, K., Stoter, J., 2025. A Methodology to Convert Highly Detailed BIM Models into 3D Geospatial Building Models at Different LoDs. *ISPRS International Journal of Geo-Information*, *14*, 465.

van Treeck, C., Rank, E., 2004. Analysis of Building Structure and Topology Based on Graph Theory. In: *Xth International Conference on Computing in Civil and Building Engineering (ICCCBE-X)*, Weimar, Germany, Bauhaus-Universität Weimar.

Van Engelenburg, C., Mostafavi, F., Kuhn, E., Jeon, Y., Franzen, M., Standfest, M., van Gemert, J., Khademi, S., 2024. MSD: A benchmark dataset for floor plan generation of building complexes. *International Archives of the Photogrammetry,* *Remote Sensing and Spatial Information Sciences*, Proceedings of the European Conference on Computer Vision Workshops.

Varghese, R., Sambath, M., 2024. YOLOv8: A Novel Object Detection Algorithm with Enhanced Performance and Robustness. *Proc. Int. Conf. Adv. Data Eng. Intell. Comput. Syst.* (ADICS).

Wang, W., Xu, K., Song, S., Bao, Y., Xiang, C., 2024. From BIM to Digital Twin in BIPV: A Review of Current Knowledge. *Sustainable Energy Technologies and Assessments*, 67, 103855.

Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M., 2019. Dynamic Graph CNN for learning on point clouds. *ACM Transactions on Graphics*, 38(5), 146:1–146:12.

Wu, X., Jiang, L., Wang, P.-S., et al., 2024. Point Transformer V3: Simpler, faster, stronger. *arXiv preprint*, arXiv:2312.10035.

klar, O., Schwab, B., Biswanath, M.K., Zhang, Q., Zhu, J., Froech, T., Heeramaglore, M., Hijazi, I., Kanna, K., Pechinger, M., et al., 2025. TUM2TWIN: Introducing the Large-Scale Multimodal Urban Digital Twin Benchmark Dataset. *arXiv preprint*, arXiv:2505.07396.

Xie, E., Wang, W., Yu, Z., Anandkumar, A., Alvarez, J.M., Luo, P., 2021. SegFormer: Simple and efficient design for semantic segmentation with transformers. *arXiv preprint*, arXiv:2105.15203.

Xu, H., Xu, J., Huang, Z., Xu, P., Huang, H., Hu, R., 2024. FRI-Net: Floorplan reconstruction via room-wise implicit representation. *arXiv preprint*, arXiv:2407.10687.

Xu, Y., Xu, W., Cheung, D., Tu, Z., 2021. Line segment detection using transformers without edges. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 4257–4266.

Yang, Q., Zhao, Y., Cheng, H., 2024. MMLF: Multi-Modal Multi-Class Late Fusion for Object Detection with Uncertainty Estimation. *arXiv preprint*, arXiv:2410.08739.

Yue, Y., Kontogianni, T., Schindler, K., Engelmann, F., 2022. RoomFormer: Single-stage room-level floor-plan reconstruction using transformers. *arXiv preprint*, arXiv:2211.15658.

Zhou, B., Zhao, H., Puig, X., Xiao, T., Fidler, S., Barriuso, A., Torralba, A., 2019. Semantic understanding of scenes through the ade20k dataset. *International Journal of Computer Vision*, *127*(3), pp.302-321.

Zhou, Y., Tuzel, O., 2017. VoxelNet: End-to-end learning for point cloud based 3D object detection. *arXiv preprint*, arXiv:1711.06396.

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

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

Appendix A

Over the three years of the PhD, a complete Scan-to-EDTs framework has been developed, culminating in a GitHub repository that will be released soon. The following section provides a detailed description of the repository.

Scan-to-EDTs GitHub Repository

Repository: *to_be_defined*

1. Creating the environment

The repository is organized so that, given a classified point cloud, it defines a Digital Twin that can then be further refined by the user according to their specific needs. The Scan-to-EDTs has been tested with Python 3.9, CUDA 12.1, PyTorch 2.2.0, torchvision 0.17.0.
The structure of the repository is as follows:

```
Scan-to-EDTs/
├── data/
├── script/
├── third_party/
├── utils/
└── README.md
```

While the subfolders' structure and the final visualization of the repository would be like:

```
Scan-to-EDTs/
├── data/
│   └── <name>.laz (point cloud)
│   └── graph_files/
│       └── <name>.ttl (created after running step_00_graph_from_laz.ipynb)
├── script/
│   ├── step_00_graph_from_laz.ipynb
│   ├── step_01_scan_to_edts.ipynb
│   ├── step_02_scan_to_edts.ipynb
│   ├── step_03_material_assignment.ipynb
│   ├── step_04_energy_simulations_uep.ipynb
│   ├── step_05_IoT_sensors_integration.ipynb
│   ├── step_06_gbxml_parser_transformer.ipynb
│   ├── step_07_ifc_parser_transformer.ipynb
│   ├── step_08_edts_module.ipynb
│   ├── context_KUL.py
│   ├── IDL_files/
│   │   └── _classes_expanded.json
│   ├── utils/
│   │   └── energyplus_utils.py
├── third_party/
└── pointcept (fork of https://github.com/Saiga1105/pointcept)
└── README.md
```

Use the environment.yml file to install all required packages. In the terminal, from the repository folder:

```
# cd Scan-to-EDTs
conda env create -f environment.yml
# activate the environment
conda activate edts
```

2. Preparing data

The scripts require the following inputs:

- A classified point cloud
- A JSON file containing the class definitions

These files allow the system to correctly interpret and map the point cloud based on the `_classes_expanded.json` file. The workflow is set up to process .laz files containing two scalar fields (SF):

1. classes (SF)

   With the following structure:

   - `Unassigned – id: 255, temp_id: -1`
   - `Floors – id: 0, temp_id: 0`
   - `Ceilings – id: 1, temp_id: 1`
   - `Walls – id: 2, temp_id: 2`
   - `Columns – id: 3, temp_id: 3`
   - `Doors – id: 4, temp_id: 4`
   - `Windows – id: 5, temp_id: 5`
   - `Lights – id: 6, temp_id: 6`
   - `Radiators – id: 7, temp_id: 7`
   - `HVAC – id: 8, temp_id: 8`

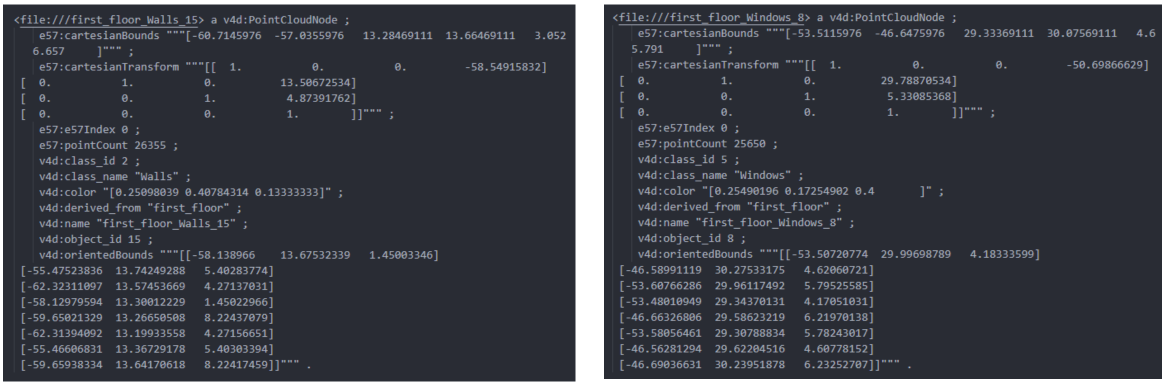
   (Each class includes color coding and description in the original file)
2. objects (SF), that defines object instances and instance-level segmentation.

3. Graph File Generation from the Point Cloud (.laz)

Once the .laz file containing RGB colors and the scalar fields classes and objects is ready, run `step_00_graph_from_laz.ipynb`.

Set the input directory where the algorithm will read the .laz files, process their scalar fields, and create the nodes. The output is a `.ttl` file structured as a graph representing the objects in the point cloud (Fig. 3). All subsequent scripts rely on this .ttl file during the reconstruction phase and use the Geomapi library (https://ku-leuven-geomatics.github.io/geomapi/index.html) for node processing.

Geomapi is already installed during environment creation (via `environment.yml`). If Geomapi fails to run, refer to its official installation instructions.



```
<file:///first_floor_Walls_15> a v4d:PointCloudNode ;
    e57:cartesianBounds """[-60.7145976  -57.0355976   13.28469111  13.66469111   3.052
   6.657     ]""" ;
    e57:cartesianTransform """[[  1.          0.          0.        -58.54915832]
 [  0.          1.          0.         13.50672534]
 [  0.          0.          1.          4.87391762]
 [  0.          0.          0.          1.        ]]""" ;
    e57:e57Index 0 ;
    e57:pointCount 26355 ;
    v4d:class_id 2 ;
    v4d:class_name "Walls" ;
    v4d:color "[0.25098039 0.40784314 0.13333333]" ;
    v4d:derived_from "first_floor" ;
    v4d:name "first_floor_Walls_15" ;
    v4d:object_id 15 ;
    v4d:orientedBounds """[[-58.138966    13.67532339   1.45003346]
 [-55.47523836  13.74249288   5.40283774]
 [-62.32311097  13.57453669   4.27137031]
 [-58.12979594  13.30012229   1.45022966]
 [-59.65021329  13.26650508   8.22437079]
 [-62.31394092  13.19933558   4.27156651]
 [-55.46606831  13.36729178   5.40303394]
 [-59.65938334  13.64170618   8.22417459]]""" .
```

```
<file:///first_floor_Windows_8> a v4d:PointCloudNode ;
    e57:cartesianBounds """[-53.5115976  -46.6475976   29.33369111  30.07569111   4.6
   5.791     ]""" ;
    e57:cartesianTransform """[[  1.          0.          0.        -50.69866629]
 [  0.          1.          0.         29.78870534]
 [  0.          0.          1.          5.33085368]
 [  0.          0.          0.          1.        ]]""" ;
    e57:e57Index 0 ;
    e57:pointCount 25650 ;
    v4d:class_id 5 ;
    v4d:class_name "Windows" ;
    v4d:color "[0.25490196 0.17254902 0.4       ]" ;
    v4d:derived_from "first_floor" ;
    v4d:name "first_floor_Windows_8" ;
    v4d:object_id 8 ;
    v4d:orientedBounds """[[-53.50720774  29.99698789   4.18333599]
 [-46.58991119  30.27533175   4.62060721]
 [-53.60766286  29.96117492   5.79525585]
 [-53.48010949  29.34370131   4.17051031]
 [-46.66326806  29.58623219   6.21970138]
 [-53.58056461  29.30788834   5.78243017]
 [-46.56281294  29.62204516   4.60778152]
 [-46.69036631  30.23951878   6.23252707]]""" .
```

Fig. 3. Structure of the .ttl file with the graph file.

4. Scan-to-Mesh Process: `step_01_scan_to_edts.ipynb`

The script requires the .laz file and its corresponding .ttl file as inputs.

It matches the .laz with the .ttl (check the paths in the Notebook) to associate the point cloud with the objects described in the .ttl and the classes defined in the .json. For each class, the code performs computational geometry operations and reconstructs both:

- a .json file containing all geometric outputs
- an .obj file containing the reconstructed geometry

In outputs > edts, a folder is created with the same name as the .laz file, containing two subfolders: one for .obj files and one for .json files, one per class.

Example, if the file name is house_01.laz, you will obtain:

house_01_walls.obj, house_01_columns.obj, house_01_openings.obj, ...

The same structure applies to the .json outputs.

This output represents a class-based solid reconstruction, describing the building and all its components in solid form.

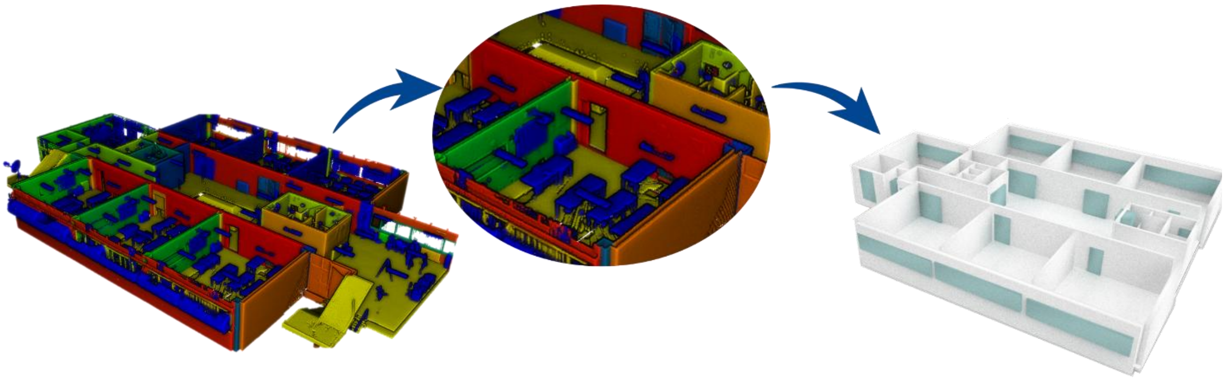


Fig. 4. Solid model derived from the RDF graph of the point cloud.

5. B-Rep Topological Model: `step_02_scan_to_edts.ipynb`
Starting from the solid outputs, the script:

- Loads the wall file (<name>_walls.json)
- Loads the wall centerlines (<name>_walls_centerline_surfaces.json)
- Projects all segments into 2D (removing the Z component)
- Iteratively computes all segment intersections
- Redefines segments when overlap with wall boundaries in the .ttl is detected
- Produces the Topologic Map
- Imports and projects openings (<name>_openings.json) into 2D
- Computes intersections between openings and wall geometry
- Reconstructs all elements in 3D, generating a topological model with manifold elements, watertight rooms, and LinearRings in walls or floors in presence of openings
- Builds the topology, hierarchy, and relationships between elements
- Saves the final customized output as a .csv file

The output is a .csv file, from which other formats and models can be derived due to its structured and hierarchical organization.

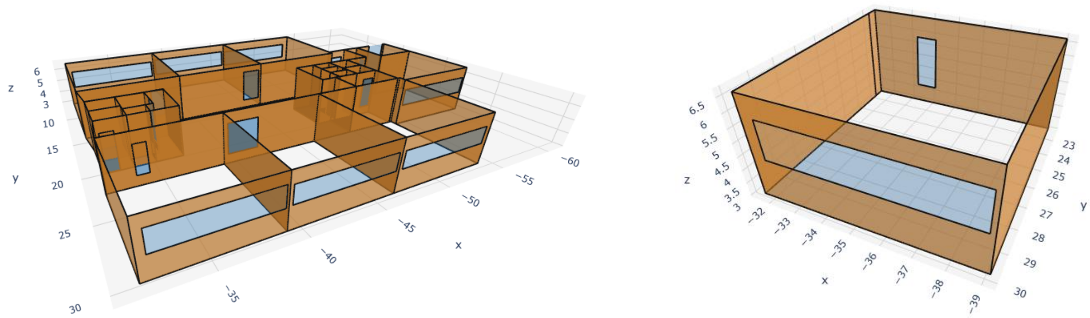


Fig. 5. Surface model. On the right, a small example showing how openings are subtracted from the surface, as required for an energy model.

Material Assignment: `step_03_material_assignment.ipynb`
The algorithm matches the ground-truth .ttl with the wall elements, assigns thicknesses as semantic attributes, and—using libraries and material definitions stored in .json files—assigns construction layers to structural elements. The .csv file from the previous step is enriched with construction packages.
Material libraries and stratigraphies are located in *script* > *ILDs* > *materials_constructions*
They can be modified for specific use cases. Results are saved in *script* > *outputs* > *jsons*

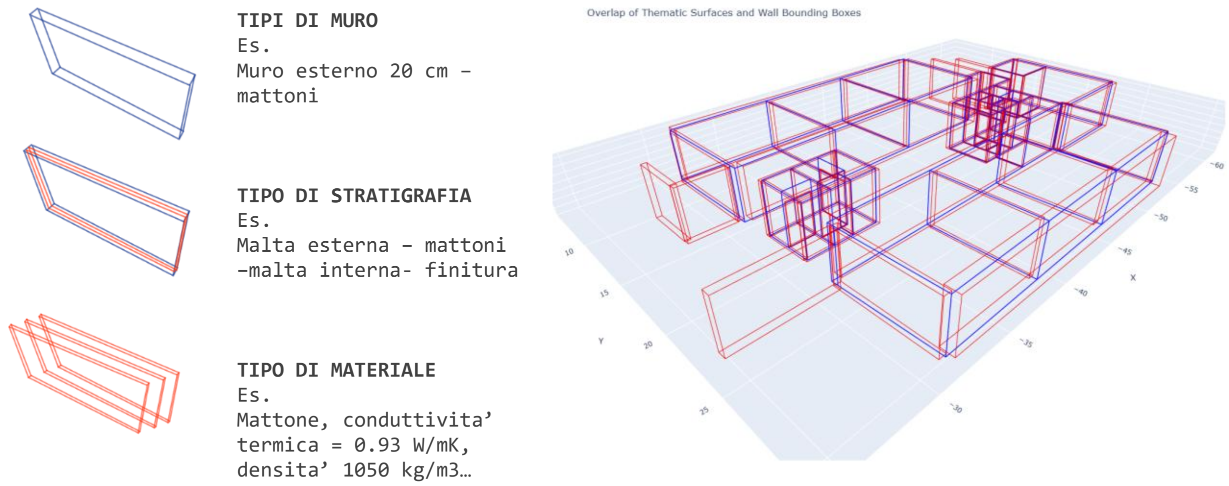


Fig. 6. Material assignment based on geospatial coordinates.

The algorithm adds columns describing the element type, its stratigraphy (layers), and the characteristics of each material layer (Fig. 7). It also computes the U-value.

	element_id	parent_id	thematic_surface	has_holes	matched_thickness	thickness_rounded	construction_name	construction_layers	u_value	geometry_wkt	element_gml_id	constru
0	3	1	external	False	0.644637	0.65	external_wall_bricks_65cm	[{'material': 'plaster_external', 'material_uu...	1.116	MULTIPOLYGON Z (((-31.666 29.983000000000004 6...	aabba56e-fdbf-48c7-869e-db25e2b21a0a	828e1dc1-5ad9-42fa-b7fe-65
1	3	1	external_with_hole	True	0.644637	0.65	external_wall_bricks_65cm	[{'material': 'plaster_external', 'material_uu...	1.116	MULTIPOLYGON Z (((-31.666 29.983000000000004 6...	aabba56e-fdbf-48c7-869e-db25e2b21a0a	828e1dc1-5ad9-42fa-b7fe-65
2	5	1	external	False	0.343848	0.35	external_wall_bricks_35cm	[{'material': 'plaster_external', 'material_uu...	2.087	MULTIPOLYGON Z (((-31.666 29.983 3.052, -31.66...	b1a65071-7ae0-46ee-943f-32a803175b10	f6170539-70aa-4b7b-a7c4-51
3	6	1	external	False	0.350000	0.35	external_wall_bricks_35cm	[{'material': 'plaster_external', 'material_uu...	2.087	MULTIPOLYGON Z (((-31.513 22.966 3.052, -31.51...	c8b75911-0a2f-4caa-a4de-ca850193fd84	f6170539-70aa-4b7b-a7c4-51
4	10	1	ceiling	False	0.250000	0.25	ceiling_concrete_25cm	[{'material': 'plaster_internal', 'material_uu...	5.265	MULTIPOLYGON Z (((-39.014 22.381 6.657, -31.50...	491dd172-46dc-4acb-a341-d9ffc339b3c9	construction_uuid_48a7fe68-cddf-4294
5	9	1	floor	False	0.300000	0.30	floor_concrete_30cm	[{'material': 'wood_parquet', 'material_uuid':...	3.228	MULTIPOLYGON Z (((-39.014 22.381 3.052, -39.11...	b96f0da8-7d93-4dc6-89e7-42129da6bf2a	construction_uuid_e3932842-7aad-4ea
6	4	3	window	True	0.050000	0.05	window_glass_5cm	[{'material': 'glass_double', 'material_uuid':...	2.285	MULTIPOLYGON Z (((-38.97 29.886 4.519, -38.97 ...	2dec6770-d4fb-42f9-b746-58cbaec7e4c0	construction_uuid_27397628-b124-4c2e
7	8	7	door	True	0.040000	0.05	door_wood_5cm	[{'material': 'wood_oak', 'material_uuid': Non...	30.000	MULTIPOLYGON Z (((-33.376 22.462 3.771, -33.37...	7c8d42cd-b5e8-4f79-b112-fba73cb303da	construction_uuid_2fb62cad-76ff-4518
8	16	11	external	False	0.644637	0.65	external_wall_bricks_65cm	[{'material': 'plaster_external', 'material_uu...	1.116	MULTIPOLYGON Z (((-39.119 29.884 6.657, -39.11...	eab703bd-a0b7-4b14-b758-4351d8288349	828e1dc1-5ad9-42fa-b7fe-65
9	16	11	external_with_hole	True	0.644637	0.65	external_wall_bricks_65cm	[{'material': 'plaster_external', 'material_uu...	1.116	MULTIPOLYGON Z (((-39.119 29.884 6.657, -39.11...	eab703bd-a0b7-4b14-b758-4351d8288349	828e1dc1-5ad9-42fa-b7fe-65
10	19	11	ceiling	False	0.250000	0.25	ceiling_concrete_25cm	[{'material': 'plaster_internal', 'material_uu...	5.265	MULTIPOLYGON Z (((-39.119 29.884 6.657, -46.39...	a8223e40-76f4-4e16-a699-c0dfca8f6ea8	construction_uuid_48a7fe68-cddf-4294
11	18	11	floor	False	0.300000	0.30	floor_concrete_30cm	[{'material': 'wood_parquet', 'material_uuid':...	3.228	MULTIPOLYGON Z (((-39.119 29.884 3.052, -39.01...	b7ec1224-51ad-4d9e-b460-c442d014a10b	construction_uuid_e3932842-7aad-4ea

Fig. 7. Colonne relative ai materiali per ciascun elemento al quale è possibile associarlo.

6. Toward Energy Simulations: creating a rich epJSON from previous outputs

Run `step_04_energy_simulations_uep.ipynb.`

It takes the file <name>_with_materials.csv from the previous step, converts it into JSON, assigns EnergyPlus-related information, and produces an EnergyPlus-compatible .json (epJSON). The outputs are saved in two .csv files:

- Indoor temperature for the entire building
- Indoor temperature per thermal zone

You can visualize temperature plots for thermal zones by plotting each CSV.

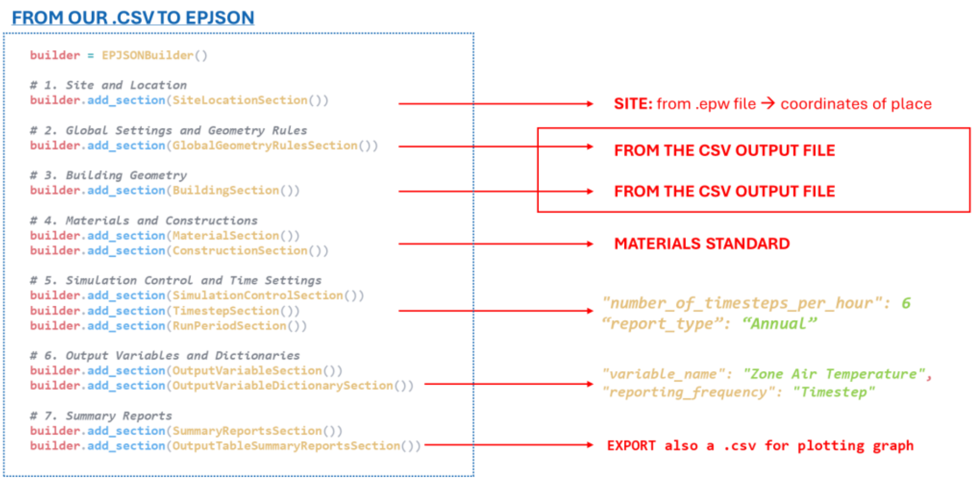


Fig. 8. Mapping the EDTS output into the epJSON file for EnergyPlus.

Module `step_05_IoT_sensors_integration.ipynb` (optional) has been integrated into `step_04_energy_simulations_uep.ipynb`.

7. Read–Parse–Transform Modules (IFC and gbXML)

- `step_06_gbxml_parser_transformer.ipynb` (Scan-to-gbXML)
- `step_07_ifc_parser_transformer.ipynb` (Scan-to-BIM)

These modules read, analyze, and transform the <name>_with_materials.csv into IFC and gbXML standard formats. This allows importing the final model into commercial tools (Revit, IDA ICE for IFC; IES VE, DesignBuilder for gbXML).

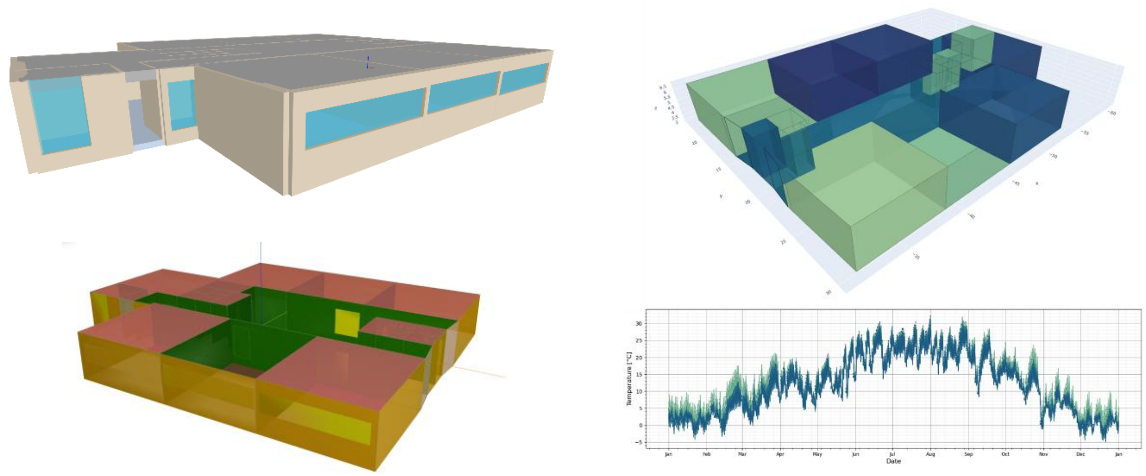


Fig. 9. Top left: IFC; bottom left: gbXML. Right: model with energy simulation results.

Both modules operate similarly: (i) they analyze the CSV and (ii) transform it into IFC or gbXML. IFC is a B-Rep solid model (like standard BIM), whereas gbXML is a surface-based B-Rep, shown as a wireframe with the same hierarchy.

8. Digital Twin Definition and Visualization

The final module, `step_08_edts_module.ipynb`, provides a 3D visualization of the surface-based B-Rep model with an HTML interface.

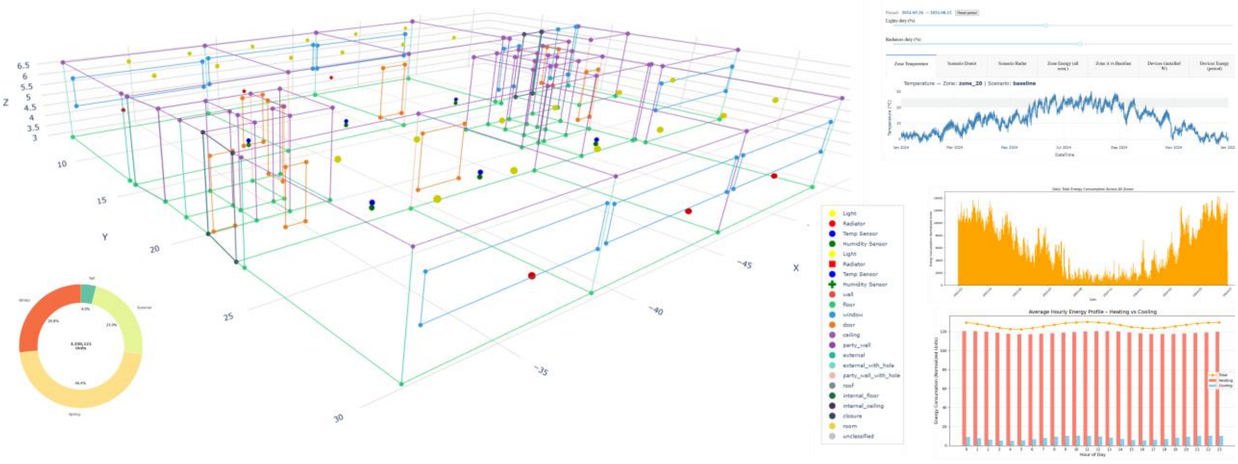


Fig. 10. Digital Twin visualization with preliminary analysis tools.

9. (Optional) Graph Visualization

To visualize the complete graph of structural and energy-related elements, run `appendix_01_graph_plotting.ipynb`.This script uses the Topologicpy library (https://topologic.app/topologicpy_doc/) to create an interactive graph visualization in PyVista (.html output).

10. (Optional) IoT Sensor Data Integration

`step_05_IoT_sensors_integration.ipynb`, incorporated into step 04, allows creating a .csv file with sensor positions and historical (or future real-time) data. Positions can be defined manually or using the sensor_detection_modules.py tool, and then imported into the main CSV.

List of Tables

List of Figures

Funding

The research was carried out within the PhD program in Industrial Innovation at the University of Trento (Italy)—38th cycle, with the support of a scholarship co-financed by Edilvi and the Italian Ministerial Decree no. 352 of 9th April 2022, based on the NRRP, CUP E66E22000050008—funded by the European Union—NextGenerationEU—Mission 4 "Education and Research", Component 2 "From Research to Business", Investment 3.3.

Data Availability Statement

Parts of the data and derived models supporting the findings of this study are available from the corresponding author upon reasonable request. Source code and processing scripts are available in the associated repository.
Raw data are available upon request to the author.

Author Contributions:
Conceptualization: Oscar Roman, Elisa Mariarosaria Farella, Fabio Remondino;
Methodology: Oscar Roman, Maarten Bassier, Giorgio Agugiaro, Ken Arroyo Ohori, Fabio Remondino;
Investigation: Maarten Bassier, Giorgio Agugiaro, Ken Arroyo Ohori;
Resources: Maarten Bassier, Giorgio Agugiaro, Ken Arroyo Ohori, Elisa Mariarosaria Farella, Fabio Remondino;
Software: Oscar Roman;
Technical Implementation: Oscar Roman, Maarten Bassier, Giorgio Agugiaro, Ken Arroyo Ohori, Fabio Remondino;
Data Curation: all authors;
Writing—Original Draft: Oscar Roman;
Writing—Review and Editing: all authors;
Formal Analysis: all authors;
Visualization: Oscar Roman;
Supervision: Maarten Bassier, Giorgio Agugiaro, Ken Arroyo Ohori, Elisa Mariarosaria Farella, Fabio Remondino;
Project Administration: Elisa Mariarosaria Farella, Fabio Remondino.

All authors have read and agreed to the published version of the manuscript.

Conflicts of Interest
The authors declare no conflicts of interest.

Publications

While the overall concept and framework have been developed in-house, components have been realized in collaboration, in particular: contributions related to Scan-to-BIM were developed at KU Leuven (Belgium), whereas the topology and semantic models were developed at TU Delft (The Netherlands). Further details have been provided in the authors' contributions section and related publications.

1. Roman, O., Bassier, M., Agugiaro, G., Arroyo Ohori, K., Farella, E. M., Remondino, F., 2025. Scan-to-EDTs: Automated Generation of Energy Digital Twins from 3D Point Clouds. *Buildings*, 15, 4060.
2. Roman, O., Farella, E. M., Remondino, F., 2025. Simulation-driven Thermal Analyses for Cultural Heritage Conservation, *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., X-M-2-2025*, 301–308.
3. Roman, O., Bassier, M., Ricciuti, S., Farella, E. M., Remondino, F., Viesi, D., 2025. Digital Twins and CFD simulations for accurate sensor positioning, *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLVIII-G-2025,* 1291–1298.
4. Roman, O., Bassier, M., De Geyter, S., De Winter, H., Farella, E. M., Remondino, F., 2024. BIM Module for Deep Learning-driven parametric IFC reconstruction*, Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLVIII-2/W8-2024,* 403–410.
5. Roman, O., Mazzacca, G., Farella, E. M., Remondino, F., Bassier, M., Agugiaro, G., 2024: Towards Automated BIM and BEM Model Generation using a B-Rep-based Method with Topological Map, *ISPRS Ann. Photogramm. Remote Sens. Spatial Inf. Sci., X-4-2024*, 287–294.
6. Roman, O., Farella, E. M., Rigon, S., Remondino, F., Ricciuti, S., Viesi, D., 2023: From 3D surveying data to BIM to BEM: the INCUBE dataset, *Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLVIII-1/W3-2023*, 175–182.
7. Roman, O., Avena, M., Farella, E. M., Remondino, F., Spanò, A., 2023: A semi-automated approach to model architectural elements in scan-to-BIM processes*, Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLVIII-M-2-2023,* 1345–1352.

Acknowledgements and thanks

Having reached the end of this long and meaningful journey, I feel the need to pause and reflect on all that it has meant to me. Over time, I have come to realize that the journey itself is the most valuable and enriching part, while the final goal is simply another step along the path of life.
Looking back, the road has been demanding, often difficult, and at times exhausting. There were moments when the weight of it all felt overwhelming, and moments when giving up seemed like the easiest choice. For this reason, first and foremost, I would like to express my deepest and most heartfelt gratitude to my family. Without their unwavering support, patience, and love, none of this would have been possible.

To my parents, Oriana and Romeo, I owe more than words can express. Thank you for always standing by me, for your endless patience, your unwavering trust, and your constant presence, both emotionally and practically. Your advice, encouragement, and sacrifices have been the solid foundation that sustained me through every challenge and allowed me to reach this point. I carry your support with me in everything I do. I also warmly thank my sister, Sara, for her constant love and support, for always being there to listen to me, for encouraging me when I doubted myself, and for sharing both the difficult moments and the small victories along the way. Your presence has made this journey lighter, and I am deeply grateful to have you by my side.

My sincere thanks go to my tutor, Professor Fabio Remondino, without whom I truly would not be where I am today. His scientific guidance, high standards, and constant encouragement to push beyond my limits have shaped not only this thesis, but also my way of thinking and working as a researcher, and more broadly, my personal growth. His insightful comments, careful corrections, and inspiring ideas have been an invaluable guide throughout the entire PhD journey.

I am equally grateful to my second tutor, Professor Elisa Mariarosaria Farella, for her constant support, availability at every stage of this work, and thoughtful feedback. Her suggestions, clarifications, and encouragement have played a fundamental role in refining and strengthening this research, and in helping me navigate the many challenges encountered along the way, both professionally and personally.
A heartfelt thank you also goes to my supervisors during my research periods abroad. I am deeply grateful to Professor Maarten Bassier, with whom I spent three months at KU Leuven learning and further developing Scan-to-BIM methodologies. That period was crucial to my understanding of the topic, and I sincerely appreciated the time, patience and insightful discussions we shared.

Finally, I would like to express my profound gratitude to Professor Giorgio Agugiaro and Professor Ken Arroyo Ohori for the remarkable three months spent working together at TU Delft. During this time, I learned immensely, both professionally and personally. Their openness, expertise and willingness to collaborate created a stimulating and inspiring environment that greatly enriched this PhD experience. Their deep knowledge of technical modelling, together with their advanced expertise in energy and semantic models, has been a constant source of inspiration and a fundamental contribution to the development of this work.

Setteville, 3 January 2026

La borsa di dottorato è stata cofinanziata con risorse dell'Unione europea-*NextGeneration EU*
Piano Nazionale di Ripresa e Resilienza (PNRR)