

Article

Sustainable Cross-Platform Reconstruction and Reuse of Semantic-Vertex-Based BIM 3D Objects

Jaeho Cho

Department of Architectural Engineering, Dankook University, Yongin-si 16890, Republic of Korea;
11991030@dankook.ac.kr

Abstract

Building Information Modeling (BIM) three-dimensional (3D) objects undergo repeated conversion and reconstruction processes for cross-platform utilization, during which geometric information loss, topological distortion, and semantic omission frequently occur, leading to fundamental limitations in accurate shape reconstruction and semantic-based functional reuse. The academic objective of this study is to overcome these limitations by proposing a three-stage sequential cross-platform reconstruction framework, consisting of semantic-vertex-based functional utilization, semantic-vertex-based invariant triangle mesh reconstruction, and semantic-vertex-based functional reuse, and to experimentally validate its effectiveness. To this end, an FBX–JSON dual-pipeline-based data management architecture is introduced to process visual geometric data and non-visual semantic metadata in parallel, thereby ensuring platform independence and data consistency. Experimental validation was conducted using IFC-based BIM objects generated in Autodesk Revit and triangle mesh models processed in Blender, at both the object and project levels. Quantitative evaluation was performed using geometric identity preservation, mesh completeness, semantic vertex restoration accuracy, and functional retention rate as the core performance indicators. The results reveal that the primary cause of mesh failure during platform transformation is face normal inconsistency, which can be stably resolved through auxiliary remeshing, thereby ensuring robust mesh reconstruction. Although the experiments were limited to round-trip transfers between Blender and Unity, the results experimentally verify the effectiveness of the proposed three-stage reconstruction framework and dual-pipeline data architecture, while also demonstrating their strong potential for generalization to broader cross-platform BIM environments.

Keywords: building information modeling (BIM); Industry Foundation Classes (IFC); semantic vertex; invariant triangle mesh; platform independence; dual-parallel pipeline

Academic Editors: Ahmad Jade, Jieying Jane Zhang and Farzad Jalaei

Received: 28 December 2025

Revised: 31 January 2026

Accepted: 6 February 2026

Published: 9 February 2026

Copyright: © 2026 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

1. Introduction

1.1. Cross-Platform BIM Degradation and Sustainability Challenges

Building Information Modeling (BIM) provides an intelligent digital representation that supports construction projects throughout their entire lifecycle [1]. BIM models consist of object-oriented 3D elements embedding geometric, material, and physical attributes [2]. However, when BIM models are exchanged across heterogeneous platforms, differences in geometric kernels, coordinate systems, and topological interpretation mechanisms frequently cause geometric distortion and semantic loss [3,4]. These inconsistencies

result in misalignment of structural elements and degradation of topological connectivity [5], thereby undermining accurate geometric reconstruction and long-term functional continuity.

Previous studies report frequent reinterpretation of vertices, edges, and faces, accompanied by distortions in rotation, scale, and coordinate origins during platform conversion [4,6], even when IFC is employed [4]. As these transformations accumulate, progressive geometric and semantic deterioration emerges, which this study defines as cross-platform BIM degradation. In practice, this degradation leads to repeated rework, redundant modeling, and regeneration of digital assets, increasing project cost and computational overhead. From a sustainability perspective, such cumulative degradation produces avoidable digital waste, including discarded intermediate models and manually corrected meshes [7].

Research Questions.

- RQ1: How can the geometric identity of BIM objects be preserved using platform-independent criteria during cross-platform conversion and reconstruction?
- RQ2: How can semantic-based functions be restored and sustainably reused after geometric reconstruction?

1.2. Limitations of Existing Exchange Formats and Interoperability Paradigms

To improve interoperability, formats such as IFC and glTF have been widely adopted [8,9]. However, IFC suffers from implementation-level limitations, including inconsistent trim-curve interpretation, shell construction strategies, and tolerance handling, which frequently produce unstable B-rep reconstructions [10]. Similarly, NURBS-based IfcBSpline-Surface representations yield divergent surface outcomes depending on tessellation strategies. Consequently, IFC cannot guarantee stable geometric reconstruction across platforms.

In practice, these inconsistencies necessitate manual re-meshing and partial remodeling, reducing efficiency and reusability. Conversely, geometry-oriented formats such as OBJ, FBX, and glTF effectively transfer visual geometry but fail to preserve semantic structures and functional capabilities, including parametric dimension updating. Prior studies report significant metadata loss and semantic degradation in FBX-based exchanges [11] and game-engine-based BIM transformations [12]. As a result, a structural dilemma arises: standardized formats compromise geometric fidelity, whereas geometry-based formats sacrifice semantic and functional integrity.

1.3. Research Gap, Objectives, and a Recovery-Oriented Perspective

This dilemma forces workflows to redefine semantic functions after geometric transfer, leading to repeated modeling and reduced lifecycle efficiency. Within the digital twin paradigm, persistent geometric identity and functional reuse across heterogeneous platforms are essential requirements [13].

Recent studies emphasize that IFC should function as a neutral intermediary rather than a final representation [3,14], highlighting the need for stable intermediate structures and platform-independent reconstruction criteria [15–20]. However, existing approaches mainly focus on geometric comparison [18] or semantic rule separation [15], without providing systematic solutions for stable geometric reconstruction and semantic-driven functional reuse. Related research in semantic web and BIM–AI integration further confirms that interoperability depends critically on semantic completeness, persistent object identity, and reliable geometry [21–23].

Nevertheless, two key challenges remain unresolved: (1) accurate reconstruction of objects with curved surfaces, openings, and complex geometries, and (2) sustained reuse of reconstructed objects for semantic-based functions, such as automatic dimension

updating [24,25]. If unresolved, repeated reconstruction cycles inevitably accumulate digital inefficiencies, undermining BIM sustainability [26].

To address these challenges, this study proposes a recovery- and reuse-oriented framework that explicitly separates platform-independent geometric reconstruction criteria from platform-independent semantic-based functional definitions, enabling stable reconstruction and consistent functional reuse across heterogeneous platforms. Rather than extending IFC, the proposed approach prioritizes stable mesh reconstruction and subsequent restoration of semantic-driven functionalities.

Definitions.

- **Reproduction:** Visual instantiation without guaranteeing geometric identity or functional editability.
- **Reconstruction:** Recovery of geometric structure and logical identity enabling functional reuse after platform transformation.

2. Related Work

Previous studies on building information modeling (BIM) interoperability have primarily focused on improving the accuracy of geometry exchange based on IFC standard or on minimizing information loss during format conversion. More recently, however, increasing attention has been paid to the long-term preservation of geometry reconstruction and reusability of BIM objects, extending beyond simple geometric exchange toward the preservation of topology, semantics, and vertex structures.

The dual-structured transformation model proposed in this study, based on functional semantic vertices, is aligned with this emerging perspective. Accordingly, related research is reviewed from four complementary viewpoints: (1) limitations inherent in the IFC standard itself, (2) IFC extension and reconversion approaches, (3) direct exchange methods beyond IFC, and (4) integration with real-time platforms such as game engines.

In particular, this section focuses on two critical issues that remain insufficiently addressed in existing studies: the lack of invariance in vertex structures and the problem of function reuse, in which object meshes must first be reliably reconstructed after platform conversion (recovery), followed by the restoration, recombination, and reuse of semantic-vertex-based functions on the recovered geometry.

2.1. IFC Data Exchange Standards and Geometry Preservation

IFC is widely adopted as the global standard for BIM data exchange; however, persistent issues related to geometric, topological, and attribute inconsistencies continue to be reported in practical cross-platform exchanges. Numerous studies have emphasized that these issues are not merely implementation errors, but stem from the intrinsic complexity and heterogeneity of the IFC representation schema itself.

Noardo et al. (2021) [24], through the GeoBIM Benchmark, demonstrated that identical IFC models are interpreted differently across BIM software platforms, leading to the loss of relationships and attributes. They explicitly stated that “issues in the standard’s implementation and use prevent its effective use”, highlighting that the diversity of IFC representations itself constitutes a fundamental cause of interoperability degradation.

Similarly, Pazlar and Turk (2008) [5], through round-trip experiments, showed that objects which appear visually identical may be internally decomposed into different B-rep structures or have their coordinate definitions reinterpreted during IFC exchanges, providing empirical evidence that IFC implementations lack platform-level consistency.

Gerbino et al. (2021) [4] demonstrated, through structural IFC model comparison, that geometry and connectivity information in IFC exchanges are subject to significant

variations even when standardized workflows are applied, indicating that the use of IFC alone does not guarantee stable preservation of geometric and topological consistency.

These limitations remain evident in recent AI- and data-driven studies. Du et al. (2024) [27] emphasized that heterogeneous geometric representations within IFC files hinder analytical consistency, concluding that such heterogeneity significantly compromises data integrity and limits the reuse of IFC-based information in advanced analytical and AI-driven applications.

Taken together, prior IFC-related studies consistently demonstrate the following:

- (1) Although IFC provides rich expressive capability, its standard framework inherently contributes to interoperability degradation;
- (2) Even when external geometry appears preserved, internal topology, reference points, and vertex structures are unstable;
- (3) Such inconsistencies accumulate during round-trip exchanges, making faithful reconstruction increasingly difficult;
- (4) These issues are systemic, arising from limitations embedded in the standard itself rather than from isolated implementation errors.

This background supports the necessity of shifting the research focus from reconstructing original IFC-native geometry to achieving sustainable recovery-optimized interoperability across platforms. Rather than extending or modifying the IFC representation itself, this perspective emphasizes the need for an independent geometric foundation capable of reliably and completely reconstructing object meshes after platform conversion, as well as a structural framework that enables semantic functions to be reused based on the recovered objects.

2.2. Studies on IFC Extension and Reconversion

To address the complexity of IFC and interpretation discrepancies across platforms, numerous studies have sought to simplify IFC representations or restructure them into alternative schemas. Rather than preserving the original IFC schema in its entirety, these approaches have increasingly evolved toward semi-IFC structures or parallel maintenance of multiple representations, tailored to specific application purposes.

Lee (2015) [15] proposed an intermediate representation (IR) to mitigate geometric information loss during shape conversion across heterogeneous BIM software environments. Similarly, Solihin et al. (2017) [16] reorganized IFC-based BIM data into a multiple-representation structure to support high-performance computation for large-scale BIM models. Yu et al. (2023) systematically reviewed recent efforts to extend IFC-based schemas, highlighting approaches for accommodating external data such as sensing information and discussing their potential relevance to digital twin-oriented applications [28].

Recent research on IFC data serialization has explored the conversion of IFC STEP files into JSON or ifcJSON formats to reduce structural complexity and facilitate efficient parsing and visualization in web-based BIM applications, potentially improving performance in lightweight and real-time processing contexts. Related studies have also investigated restructuring IFC data into GIS-oriented formats, such as CityGML and CityJSON. Zhu et al. [29] analyzed the IFC-to-CityGML conversion process and reported geometry simplification, vertex remapping, and attribute omission arising from differences in levels of detail (LOD), thereby highlighting the structural limitations of conversion-based approaches for BIM–GIS integration.

Collectively, these IFC extension and reconversion studies are closely related to the strategy adopted in this research, which maintains visual geometry representations (e.g., FBX and OBJ) in parallel with non-visual structural representations (e.g., JSON-based semantic and vertex structures). A common conclusion across these studies is that a single,

monolithic IFC structure is insufficient to guarantee long-term reusability; instead, structural information must be separated into dedicated layers or stored concurrently in lightweight formats according to application requirements.

2.3. Studies on Direct Cross-Platform Exchange Approaches

An increasing number of studies have explored direct cross-platform exchange strategies that reduce reliance on monolithic IFC workflows by combining visual-based formats (e.g., FBX, OBJ, glTF, and Universal Scene Description (USD)) with parallel non-visual structural representations (e.g., JSON/ifcJSON, CityJSON, XML, or USD meta-layers). These approaches aim to improve interoperability by leveraging lightweight geometry for visualization alongside complementary non-visual data structures for semantics and structure [30,31]. Nam et al. (2019) [32] demonstrated that lightweight 3D formats, such as glTF, provide superior data delivery efficiency and real-time rendering performance than IFC in mobile web browser environments, highlighting their suitability for web-based visualization applications.

Du et al. (2020) reported that lightweight visual formats such as OBJ and FBX provide more stable and efficient performance for mobile BIM visualization, making them more suitable for mobile and web-based environments than directly handling IFC models [33].

Technical documentation and industry reports on USD have emphasized its ability to enable high-performance data interchange among heterogeneous tools while preserving scene structure through non-destructive composition mechanisms [34,35].

Despite these advantages, direct exchange approaches exhibit clear structural limitations, as visual formats often lack mechanisms for semantic and relational preservation [12]. Biljecki et al. (2016) [36], Banfi (2021) [37], and Hakim et al. (2024) [38] similarly documented issues related to semantic information loss, topological inconsistencies, and level-of-detail (LOD) mismatches across BIM–GIS and immersive visualization workflows.

In summary, while direct exchange methods offer superior performance for real-time visualization, they remain insufficient for the stable preservation of vertex structures, topology, and reference points required for analysis, validation, and two-way (round-trip) integration. This limitation provides a clear rationale for the adoption of a parallel non-visual structure based on semantic vertices stored in structured JSON data. Furthermore, in this study, stability based solely on visual reproduction is insufficient; instead, complete and reliable reconstruction of the object mesh must precede platform transformation, after which functional reuse is performed on the reconstructed object.

2.4. BIM Integration with Game Engines and Real-Time Platforms

As BIM applications expand into real-time platforms such as VR, AR, visualization, education, safety training, and 4D simulation, extensive research has explored integrating BIM with game engines (e.g., Unity and Unreal Engine). However, most existing workflows rely on multi-stage conversion pipelines (e.g., BIM → IFC → FBX/OBJ/glTF → engine), during which parametric attributes, relational structures, and coordinate system information may be degraded or lost. Consequently, prior studies have noted the need for substantial post-processing after mesh conversion to adapt BIM data for real-time engine environments (May et al., 2022 [39]; Johansson & Roupé, 2024 [40]; Zaman et al., 2024 [41]; Wang et al., 2025 [42]).

Rehman et al. (2025) [12], Afzal et al. (2021) [43], and Babalola et al. (2023) [44] collectively indicate that many engine-based BIM integrations remain largely one-way, focusing on visual mesh transfer rather than bidirectional or semantically rich exchange. Consequently, existing approaches that do not explicitly separate visual meshes from

parametric and semantic structures face limitations in supporting sustainable reconstruction and cross-platform reusability.

Accordingly, the dual-parallel structure proposed in this study, which maintains visual meshes alongside a semantic-vertex-based non-visual structure (JSON), can serve as an effective alternative for preserving information consistency between BIM and real-time engines. In particular, within the context of this research, “reproduction” in a game engine refers solely to visual instantiation and does not guarantee object identity. Therefore, the core challenge in engine-based utilization lies in ensuring stable reconstruction of object meshes after platform transformation, followed by the subsequent recombination, updating, and reuse of semantic-vertex-based functions on the recovered object.

2.5. Synthesis of Research Directions

Based on a comprehensive review of related studies, several structural conclusions consistently emerge. First, a single IFC-based representation has been shown to be insufficient for the stable preservation of geometric, topological, and semantic information across heterogeneous platforms. Recent studies on IFC extension, reversion, and topological abstraction indicate that these limitations are not merely implementation-related but are rooted in the geometry-centric coupling of representation and semantics within the IFC schema, thereby highlighting the necessity of separating geometric representations from semantic and relational structures [45].

In parallel, research on lightweight BIM transformation and web-based visualization demonstrates that direct exchange approaches—such as WebGL-oriented lightweighting pipelines—can significantly improve real-time performance and accessibility. However, these approaches typically prioritize rendering efficiency, which results in reduced semantic richness, weakened object identity, and limited topological consistency, thereby constraining their applicability for functionally sustainable and reusable BIM workflows [46].

Furthermore, multiple studies involving game-engine-based BIM integration report recurrent losses of vertex structures, coordinate consistency, and parametric information during platform transitions, confirming that visual geometry alone cannot serve as a reliable reference for object reconstruction or functional reuse.

Consequently, the dual-parallel architecture proposed in this study—combining visual formats (visual meshes) with semantic-vertex-based non-visual structures (semantic JSON)—emerges as a practical solution for achieving cross-platform invariance of vertices, topology, reference points, and dimensional information that existing studies have not adequately addressed. Notably, this approach is fundamentally distinguished from prior research in that it presupposes an invariant triangle mesh as the geometric foundation for complete and prioritized object reconstruction, upon which semantic-vertex-based functions are subsequently restored, recombined, and continuously reused based on the recovered object.

3. A New Perspective

Research on BIM interoperability has long been centered on the industry foundation classes (IFC). However, in practical cross-platform exchanges, IFC-based geometry and topology are frequently not preserved faithfully due to differences in interpretation among software platforms. Although IFC provides a highly expressive representation, variations in topological rules and internal geometric kernels across BIM authoring tools and visualization or analysis platforms commonly lead to vertex restructuring, geometric distortion, and loss of attributes.

To address these issues, many studies have attempted to extend or restructure IFC. Nevertheless, such approaches remain fundamentally limited in restoring objects to the parametric level of each platform's native geometric engine.

More recently, attempts have emerged to decouple visual formats (e.g., FBX, OBJ, glTF, and USD) from non-visual structural representations (e.g., JSON, XML, CityJSON, and IFC-SPF). Mesh-based visual formats enable efficient rendering but do not preserve semantic and topological structures, whereas non-visual formats effectively store semantic information but often fail to reconstruct stable meshes, leading to inconsistent vertex indices, reference points, and geometric deformation. Consequently, repeated platform transitions cause cumulative degradation due to vertex reordering, coordinate transformations, and LOD simplification. These problems become more severe in game-engine-based environments, where BIM models undergo multi-stage conversion pipelines (e.g., BIM → IFC → FBX/OBJ/glTF → engine). Throughout this process, parametric objects are reduced to static meshes, resulting in geometric distortion, coordinate inconsistencies, object fragmentation, and the practical infeasibility of restoring original BIM representations.

Against this background, the present study does not aim to regenerate native parametric BIM objects. Instead, recovery is redefined as the prioritized and stable reconstruction of the parent object mesh. The proposed approach introduces an invariant triangle-mesh-based recovery strategy, enabling logical restoration of geometric form and topological structure and direct execution within native geometric engines.

The core contribution lies in maintaining invariant logical geometry while enabling functional reuse of semantic vertices. To achieve this, a dual-structured data model is proposed, in which visual meshes and non-visual semantic-topological data are independently managed. Visual formats (FBX, OBJ, glTF) restore and execute parent object meshes, while non-visual formats (JSON, XML) store semantic vertices, vertex indices, reference points, and topological relationships as platform-independent logical structures. This dual representation preserves object identity and functional continuity across platform transitions. Through this architecture, recovered meshes maintain geometric stability under rotation, scaling, and coordinate transformations, while semantic-vertex-based functions—such as automatic dimensioning, object editing, and VR-based alignment—are consistently reproduced within native geometric engines.

Logical Reconstruction Criteria

Unlike conventional dual-structure approaches (e.g., Solihin et al. 2017) [16], which mainly reorganize geometric and semantic data for computational efficiency, this study defines explicit logical reconstruction criteria for stable recovery, preservation, and reuse across heterogeneous platforms. By treating invariant triangle meshes and semantic vertices as logical anchors, persistent object identity, geometric stability, and functional continuity are ensured. Consequently, BIM objects are no longer treated as static conversion results but as logically recoverable and functionally reusable entities that support repeated reconstruction, editing, updating, and cross-platform reuse.

Figure 1 summarizes this logical reconstruction framework, illustrating the dual-parallel pipeline for parent object mesh recovery, logical topological restoration, and semantic-based functional regeneration.

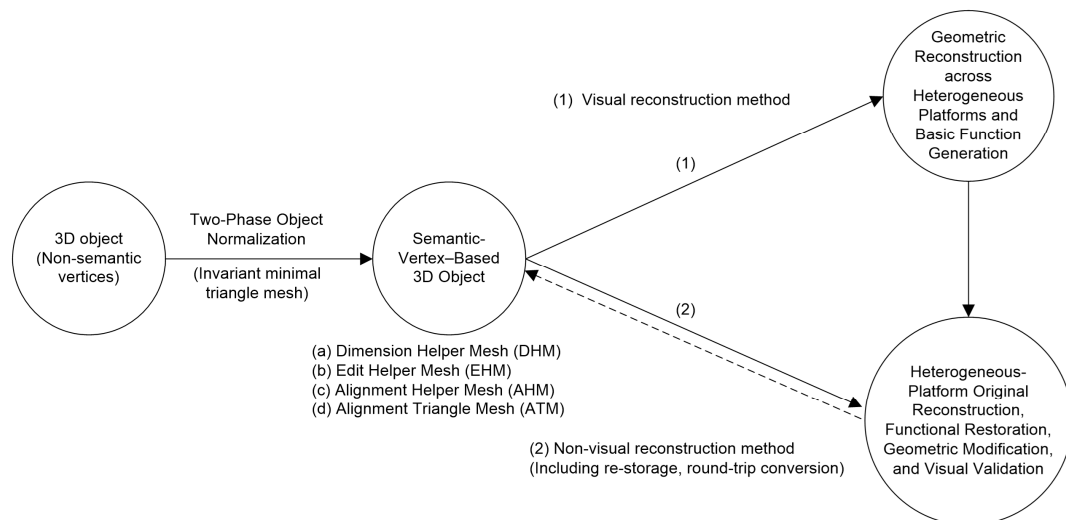


Figure 1. Conceptual framework for visual, topological, and semantic functional restoration of BIM objects.

4. Semantic Vertices and Semantic-Vertex-Based BIM 3D Objects

Functional semantic vertices refer to a subset of object vertices that play a direct geometric or functional role in defining the dimensions, spatial boundaries, or reference features of BIM objects. These vertices are selected based on stable geometric-topological criteria, independent of modeling styles or software-specific representations. For structural elements, functional semantic vertices typically include vertices defining load-bearing boundaries, section extremities, and reference axes. For architectural elements, they include vertices defining spatial limits, openings, and dimensional control points. These semantic vertex selection principles were systematically established and validated through a semantic vertex detection algorithm for dimension measurement in the author's previous study [47]. In the present study, they are briefly summarized with a focus on reconstruction and functional reuse.

In this study, semantic vertices are redefined not as auxiliary geometric points, but as intrinsic functional reference units that structurally govern object-level behaviors such as dimension generation, geometric editing, and spatial alignment. Accordingly, BIM 3D objects are treated not as mere visual meshes, but as semantic-vertex-based functional structures whose operational capabilities are preserved across heterogeneous platforms.

Unlike conventional direct-exchange approaches (e.g., FBX, OBJ, STL, glTF), which support only visual geometry transfer, semantic-vertex-based objects explicitly encode functional references, enabling post-conversion recovery, reuse, and extension of object-level functions. Thus, recovery in this research denotes the reconstruction of BIM objects as functional units, rather than the reproduction of visual appearance alone. Implementation strategies may vary across platforms and engines.

4.1. Semantic-Vertex-Based Object Structure

Semantic vertices are implemented through helper meshes, which provide executable representations of functional reference vertices without modifying the original object geometry. This structural separation allows functional logic to remain stable during platform transitions.

Three functional helper meshes are defined:

- Dimension Helper Mesh (DHM): supports dimension generation and visualization.
- Edit Helper Mesh (EHM): provides one-to-one correspondence with invariant triangle mesh vertices for geometric editing.

- Alignment Helper Mesh (AHM): enables positional and orientational alignment with external data, optionally forming an alignment triangle mesh (ATM).

All helper meshes are strictly anchored to the local coordinate system of the parent object, ensuring that semantic vertices and their associated functions remain invariant under translation, rotation, scaling, and cross-platform transformation.

4.2. Relationship to Previous Semantic Vertex Studies

Previous studies have employed semantic vertices mainly for localized tasks such as automated dimensioning [47]. In contrast, this study generalizes semantic vertices as expandable functional units, introducing a platform-independent structural framework that enables cross-platform object recovery, geometric editing, alignment, and repeated round-trip transformations. This expansion establishes semantic vertices as the core structural elements governing functional continuity of BIM objects.

4.3. Object Normalization for Semantic Vertex Application

To ensure robustness and consistency, IFC objects undergo a two-stage normalization:

- Geometric normalization: conversion to invariant triangle meshes.
- User-perspective normalization: reference-based coordinate realignment.

Based on the normalized geometry, semantic vertices are classified into dimension, edit, and alignment categories, forming a unified functional reference system that supports extensible downstream applications.

5. Integrated Framework

To address the inherent limitations of cross-platform BIM data transfer, this study proposes an integrated object reconstruction framework centered on semantic vertices and dual-structured data management. Semantic vertices provide invariant logical reference points that preserve functional meanings independent of platform-specific geometric representations, while function-specific helper meshes (DHM/EHM/AHM) operationalize these references for practical execution. By decoupling visual mesh data from non-visual logical structures, the framework enables stable object recovery and functional continuity even under geometric deformation, topological variation, and repeated platform migration. As illustrated in Figure 2, the proposed method enables cross-platform object reconstruction by first stabilizing geometry through minimum-unit invariant triangle mesh normalization, followed by the recovery of semantic-vertex-based functional reference structures and their subsequent functional reuse. This mechanism conceptually demonstrates how geometric integrity and functional continuity can be jointly preserved between heterogeneous environments such as Blender and Unity.

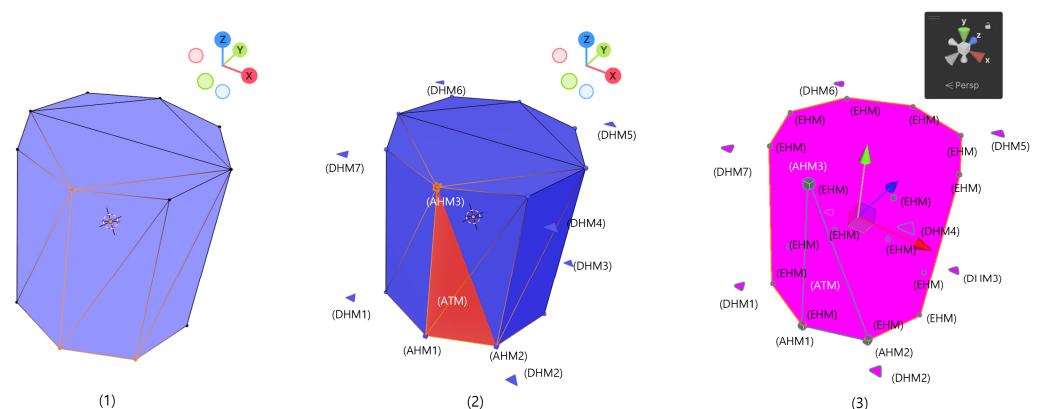


Figure 2. Cross-platform reconstruction via invariant triangle mesh normalization and semantic-vertex-based functional reuse. (1) Invariant minimal-unit triangle mesh normalization on the Blender platform; (2) Generation of semantic vertex helper meshes (DHM, EHM, AHM, ATM) on an invariant triangle mesh and (3) Cross-platform object reconstruction in Unity based on a JSON-reconstructed object (shown in magenta), enabling reuse of semantic vertex helper meshes on an invariant triangle mesh.

The overall reconstruction and reuse process is organized as an eight-stage closed-loop pipeline, as shown in Figure 3, forming a round-trip-capable workflow that systematically maintains geometric identity, semantic consistency, and functional continuity under repeated cross-platform transfers. The eight stages are summarized as follows:

1. **Pre-Recovery Optimization:** Diverse geometric representations are normalized into invariant triangulated meshes to ensure topological stability and reliable semantic vertex detection.
2. **Semantic Vertex Generation:** Semantic vertices are defined as invariant logical reference elements that preserve consistent functional meanings across heterogeneous platforms.
3. **Semantic-Based Functional Execution:** Dimensioning, editing, and alignment functions are executed based on semantic vertices to ensure platform-independent functional consistency.
4. **Dual-Structured Data Storage:** Visual mesh data and non-visual semantic reference data are separately stored and cross-linked, enabling robust recovery despite geometric distortions.
5. **Loading and Recovery:** During object loading, semantic reference structures are concurrently restored with visual rendering, enabling functional reconstruction beyond mesh reproduction.
6. **Semantic-Based Reuse:** Recovered semantic references enable repeated reuse of functional operations without redefining reference criteria.
7. **Object Transformation and Updating:** Semantic vertices and helper meshes are dynamically updated to reflect geometric modifications, maintaining synchronization between geometry and functionality.
8. **Round-Trip Saving and Recovery:** Updated objects are resaved in dual-structured formats, enabling stable cyclic recovery under repeated cross-platform transfers.

Through this integrated and cyclic framework, the proposed approach systematically alleviates geometric distortion, reference loss, and functional discontinuity, providing a practical and scalable solution for sustainable BIM object reconstruction, functional reuse, and interoperability in heterogeneous software environments.

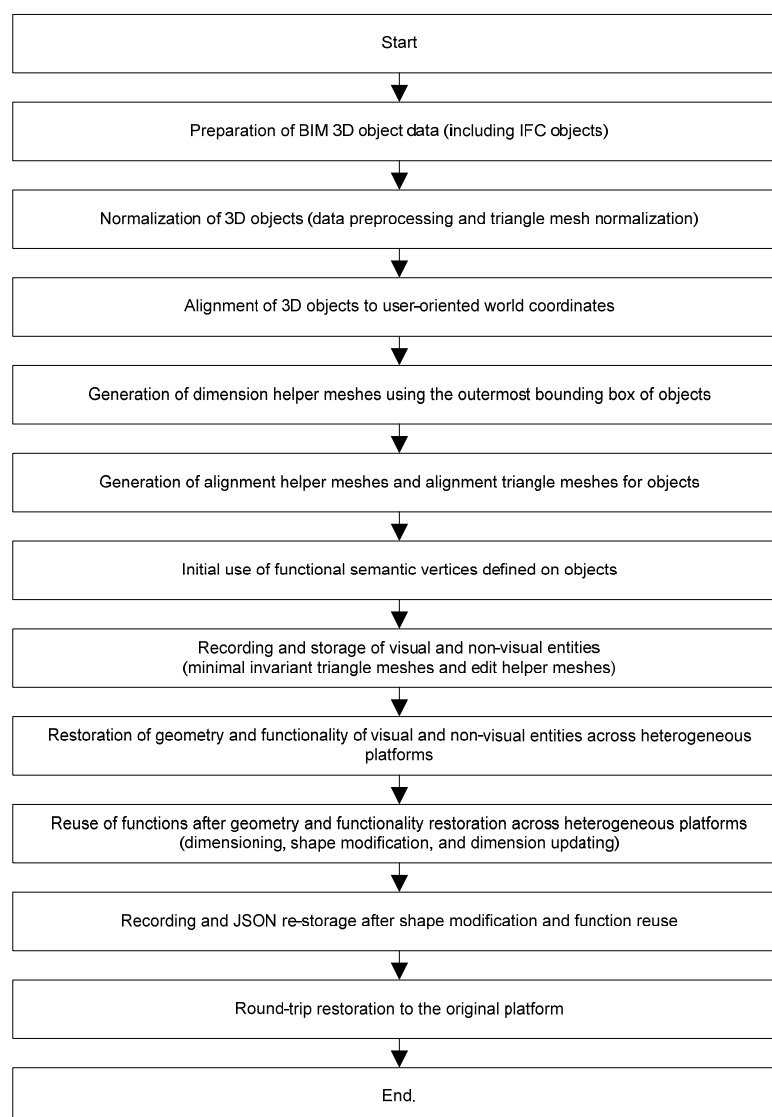


Figure 3. Overall workflow of 3D object reconstruction, reuse, and round-trip framework.

6. Integrated Dual-Structured Pipeline for Storage, Recovery, and Functional Reuse

This section presents an integrated dual-structured pipeline designed to simultaneously achieve efficient visual reconstruction, stable logical recovery, and persistent functional reuse under repeated cross-platform BIM transformations. By explicitly decoupling visual representation from semantic-based logical reconstruction while preserving semantic coherence through invariant reference structures, the proposed pipeline provides a practical implementation architecture that systematically balances reconstruction accuracy, functional robustness, computational efficiency, and storage scalability.

6.1. Conceptual Overview of the Dual-Structured Pipeline

The proposed pipeline consists of two complementary entities: the Visualization Recovery Entity (VRE) and the Non-Visualization Recovery Entity (NVRE). These two layers are defined as logically interconnected yet operationally independent structures, enabling flexible configurations ranging from lightweight visualization-oriented reconstruction to fully semantic-based object regeneration.

Here, the term dual structure does not imply mandatory parallel generation, storage, or activation of both entities. Instead, it denotes an adaptive architectural paradigm in

which the two layers may be selectively activated, coupled, or decoupled depending on object complexity, application objectives, and computational resource constraints. This selective coupling strategy ensures computational efficiency for large-scale BIM data while simultaneously supporting sustainable functional reuse.

As illustrated in Figure 4, the proposed dual-structured pipeline integrates visual reconstruction (VRE) and semantic-based logical recovery (NVRE) into a unified yet flexibly coupled architecture. In this study, a JSON-based logical storage structure is adopted for platform independence and implementation simplicity. However, the proposed semantic-vertex-based reconstruction framework is not restricted to JSON representation and can be naturally extended to various binary, hierarchical, and high-performance data formats, such as HDF5, Protocol Buffers, FlatBuffers, Apache Arrow, and USD, thereby enabling scalable, efficient, and real-time reconstruction for large-scale BIM and digital twin environments.

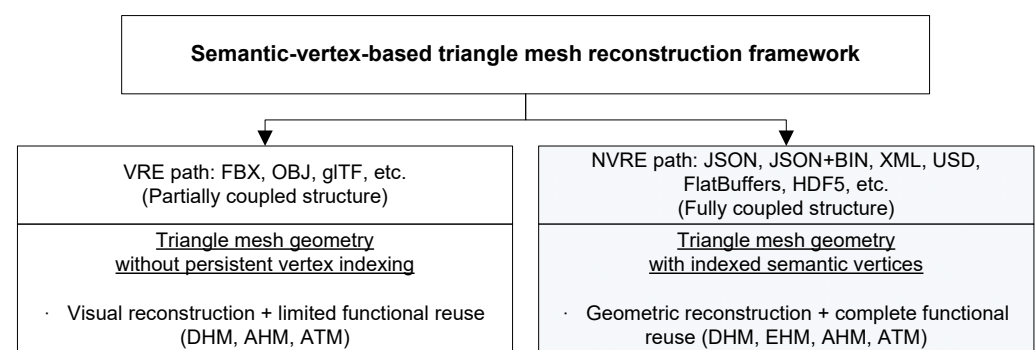


Figure 4. Conceptual architecture of the proposed dual-structured pipeline for visual reconstruction and semantic-vertex-based logical recovery.

6.2. Visualization Recovery Entity (VRE)

The VRE constitutes a visual-centric reconstruction layer based on mesh-oriented formats such as FBX, OBJ, and glTF, enabling immediate rendering and spatial cognition in heterogeneous platforms including Unity, Unreal Engine, and Twinmotion. Its primary objectives are real-time visualization, spatial comprehension, visual verification, and immersive environment construction.

VRE encapsulates triangulated geometry, material properties, transformation parameters, and hierarchical relationships to ensure accurate spatial reproduction across platforms. Optional inclusion of Dimension Helper Meshes (DHMs) allows persistent visualization of dimensional information. In contrast, semantic relations, editing rules, and logical reconstruction mechanisms are deliberately excluded, confining VRE strictly to lightweight visual representation.

6.3. Non-Visualization Recovery Entity (NVRE): Invariant Triangle Mesh-Based Logical Reconstruction and Storage

The NVRE represents the core logical reconstruction layer, responsible for preserving object identity, semantic consistency, and functional continuity. An illustrative example of VRE-based JSON usage is provided in Appendix A to demonstrate how NVRE information can be structured, stored, and reused in practice. Its foundation is the invariant triangle mesh, a platform-independent geometric reference structure in which all vertices are assigned unique identifiers, and each triangle is represented as a triplet of vertex IDs. This design ensures stable geometric and topological identity despite platform-induced retriangulation, vertex reordering, and coordinate system transformations.

Logical reconstruction proceeds sequentially through restoration of the invariant triangle mesh → recovery of semantic-vertex-based helper meshes, ensuring that semantic structures and functional reference systems are always rebuilt upon a stable geometric foundation. This reconstruction order fundamentally eliminates cumulative geometric distortion and semantic degradation under repeated cross-platform transformations.

All logical information in the NVRE is stored using a JSON-based logical container, which includes vertex IDs and coordinates, triangle compositions, and structured helper meshes, including the Dimension Helper Mesh (DHM), Editing Helper Mesh (EHM), Alignment Helper Mesh (AHM), and Alignment Triangle Mesh (ATM). All references are defined based on vertex IDs rather than absolute coordinates, enabling robust logical reconstruction regardless of mesh regeneration or coordinate transformation.

Furthermore, the framework supports an alternative selective storage strategy, in which only semantically significant vertices are preserved, enabling efficient control of storage volume and computational cost even for large-scale BIM models. This strategy ensures practical scalability for high-resolution meshes and free-form surface-based objects.

6.4. Helper-Mesh-Based Functional Reuse Framework

Functional reuse is realized through semantic-vertex-based helper mesh structures defined upon the invariant triangle mesh, transforming BIM objects from visually defined entities into function-centric computational objects.

- The Dimension Helper Mesh (DHM) preserves dimensional semantics using vertex-ID-based reference relationships, enabling immediate dimension recovery and reuse without recomputation.
- The Editing Helper Mesh (EHM) establishes a one-to-one mapping with invariant mesh vertices, providing a platform-independent interface for consistent geometric editing.
- The Alignment Helper Mesh (AHM) and Alignment Triangle Mesh (ATM) encode semantic alignment criteria and their geometric realizations, ensuring stable object correspondence across heterogeneous platforms and coordinate systems.

Together, these helper meshes enable persistent dimensioning, editing, and alignment operations under repeated cross-platform transformations, thereby guaranteeing sustainable functional continuity.

7. Experimental Results of Blender–Unity-Based Reconstruction and Functional Reuse

7.1. Fundamental Validation Using a Blender-Based Mesh Mode

This section presents a fundamental validation of the proposed semantic-vertex-based reconstruction framework using a Blender-based mesh model. At this stage, IFC models are intentionally excluded in order to establish a minimal and independent experimental baseline. Prior to IFC-based round-trip validation, this experiment aims to verify the core components of the proposed framework, including mesh preprocessing, triangular mesh normalization, semantic vertex generation, and dual-structured reconstruction.

Figure 5 illustrates the preprocessing and normalization pipeline for generating semantic-vertex-based 3D object representations. In Figure 5(1), the raw mesh object generated in Blender undergoes geometric preprocessing, including the removal of degenerate faces, elimination of redundant vertices, and mesh structure refinement, to establish a stable geometric foundation. In Figure 5(2), triangular mesh normalization is applied to generate a standardized and invariant triangular representation, ensuring consistent vertex correspondence after platform transformation. Finally, in Figure 5(3), user-oriented

alignment and semantic vertex generation are performed to establish functional reference structures, including the Dimension Helper Mesh (DHM), Alignment Helper Mesh (AHM), and Axis Transformation Mesh (ATM), followed by the first operational application of DHM-based dimensioning, resulting in a semantic-vertex-based 3D object representation that enables functional reuse.

Figure 6 compares three reconstruction strategies implemented in Unity. Figure 6(1) shows the FBX-based visual reconstruction (VRE), which verifies the preservation of geometric shape and surface continuity in a real-time rendering environment. Figure 6(2) presents the parallel dual reconstruction, in which FBX-based visual reconstruction and JSON-based non-visual reconstruction (NVRE) are simultaneously applied and spatially overlaid to directly validate geometric consistency and reconstruction accuracy. Figure 6(3) demonstrates the pure JSON-based non-visual reconstruction, in which triangle meshes and semantic vertices (DHM, EHM, AHM) are restored independently of visualization formats. For improved vertex readability and analytical clarity, the reconstructed object is displayed using a projected mesh configuration.

This baseline experiment confirms that the proposed framework supports independent visual reconstruction, pure semantic reconstruction, and their parallel operation, thereby establishing the fundamental feasibility and robustness of the dual-structured reconstruction pipeline.

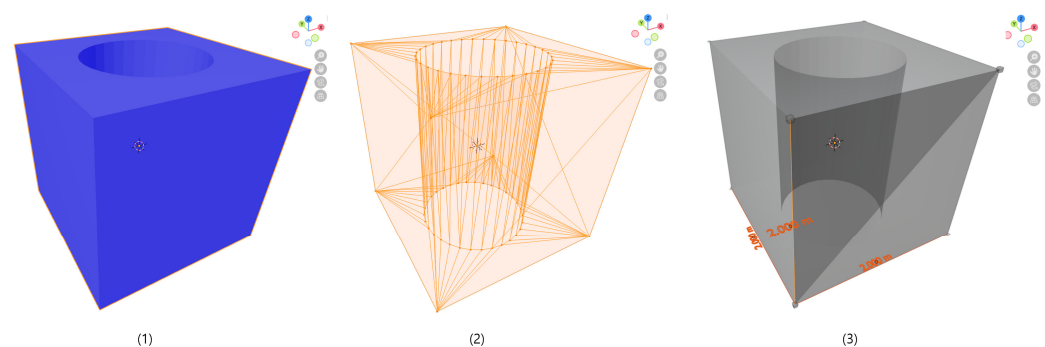


Figure 5. Preprocessing and normalization pipeline for semantic vertex initialization and first application of DHM-based dimensioning in Blender. (1) Preprocessing of mesh-based object data in Blender (the blue mesh indicates face normal directions); (2) Normalization of a triangular mesh into a transparent projected mesh representation; (3) Semantic-vertex-based 3D object representation and initial application of DHM-based dimensioning.

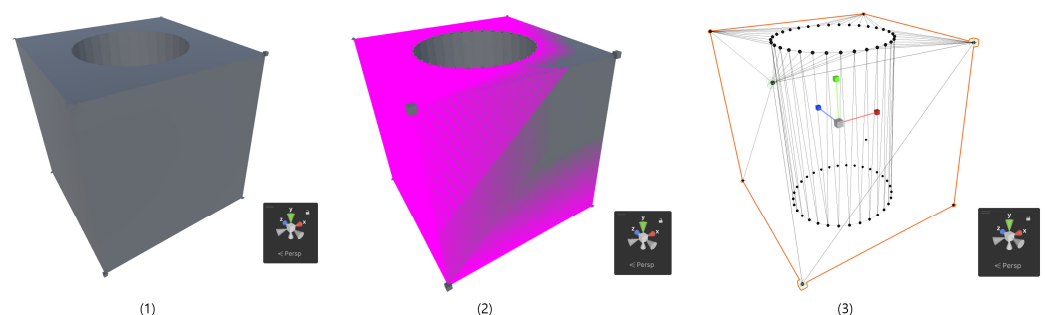


Figure 6. Comparison of FBX-based VRE, parallel dual reconstruction, and pure JSON-based NVRE. (1) VRE: FBX-based visual reconstruction in Unity, where the reconstructed mesh is rendered in gray, enabling functional reuse of semantic vertices (DHM); (2) VRE + NVRE: FBX-based visual reconstruction (gray mesh) integrated with JSON-based non-visual reconstruction (overlaid magenta mesh) in Unity; (3) NVRE: JSON-based reconstruction of triangle meshes and semantic vertices (DHM, EHM, AHM, and ATM) performed in Unity using a transparent projected mesh representation.

7.2. Shape Modification After NVRE and Reuse Function in Semantic Vertex

Figure 7 presents the results of shape regeneration through the manipulation of the Edit Helper Mesh (EHM) after JSON-based non-visual reconstruction (NVRE). In this experiment, the object was reconstructed solely from semantic vertex data and triangular mesh information stored in JSON, without relying on any visual exchange format. Subsequently, the EHM associated with the reconstructed object was arbitrarily manipulated by the user in the target platform to validate geometric editability. Specifically, the vertices of the upper circular region were intentionally translated in the vertical direction, and the cube geometry was regenerated by reflecting the modified vertex configuration. The experimental results demonstrate that the proposed semantic-vertex-based framework enables stable shape modification and mesh regeneration even after non-visual reconstruction, confirming that object geometry can be directly edited and consistently updated across heterogeneous platform environments.

In addition, Figure 7(1) illustrates the continuous reuse of the dimension generation functionality in the object reconstructed via JSON-based non-visual restoration. Here, “reuse” does not indicate the creation of new dimensions after reconstruction; rather, it refers to the preservation and continued utilization of the dimension generation logic and functional reference structures that were already defined and applied in the source platform prior to reconstruction, thereby maintaining the same logical rules and reference coordinate system. In other words, the functional reference system required for dimension generation is restored together with the object’s geometric information through the JSON representation.

Figure 7(2) further demonstrates that, even after geometric editing using the EHM, the dimension values are automatically updated based on the modified coordinates through the Dimension Helper Mesh (DHM). This confirms that dimension generation and visualization functionalities remain continuous and uninterrupted, even after post-reconstruction shape modifications.

These experimental results demonstrate that the proposed semantic-vertex-based object reconstruction framework extends beyond mere geometric restoration and ensures stable functional continuity and reusability—including dimension generation—even after shape modification.

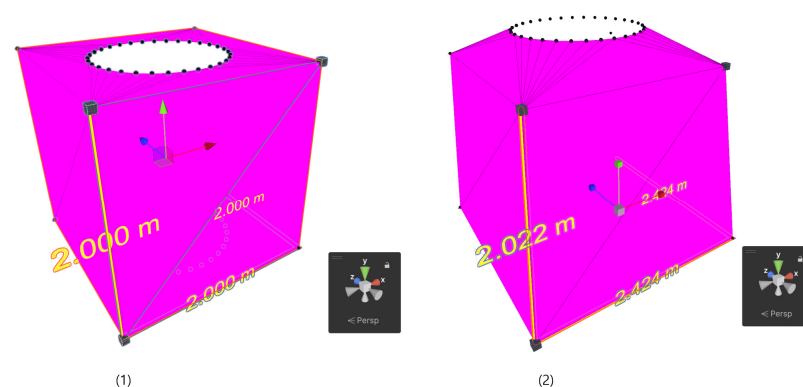


Figure 7. Dimension reuse and automatic update after NVRE-based reconstruction and geometric editing. (1) NVRE + DHM: Reuse of DHM-based dimensions on a Unity-based NVRE (magenta) mesh; (2) EHM editing: Automatic dimension update on a Unity-based NVRE (magenta) mesh.

7.3. Object Alignment Using Alignment Helper Meshes (AHMs)

Figure 8 presents a representative application example in which an alignment triangle mesh is generated using three Alignment Helper Meshes (AHMs), reconstructed on the Unity platform, and applied to object alignment with an external photograph. The

alignment triangle mesh is derived from three AHMs defined as semantic alignment references, and both the helper meshes and the alignment triangle mesh are consistently restored across heterogeneous platforms.

External object alignment, as demonstrated in this example, is a core technique required in recent AR/VR/MR applications. In this context, the present case illustrates the fundamental role of alignment helper meshes as semantic reference structures for accurate and reliable object alignment. This study does not aim to address the detailed implementation of alignment algorithms; instead, it emphasizes the conceptual and functional introduction of alignment helper meshes as semantic vertices within the proposed framework. After object alignment, dimension visualization and geometric projection are shown to be feasible, confirming that aligned objects can support subsequent functional operations. This example verifies that alignment helper meshes, defined as semantic vertices, enable not only cross-platform reconstruction but also practical alignment-based reuse scenarios while preserving functional continuity.

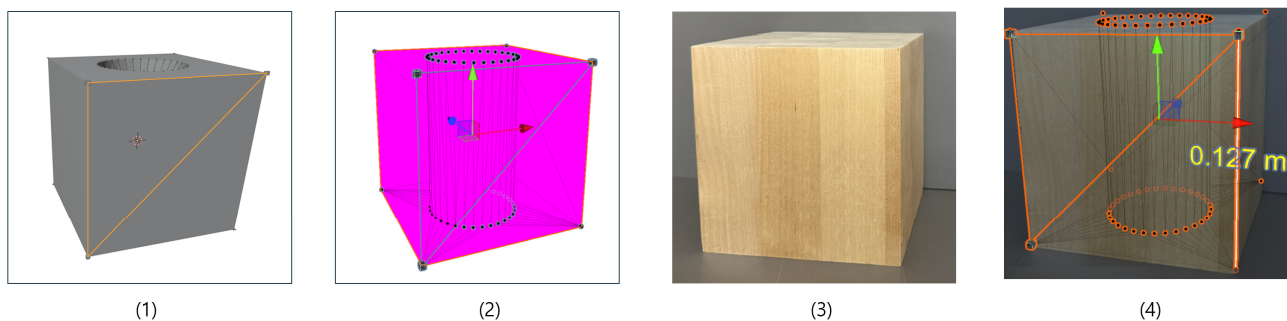


Figure 8. Object alignment using AHM across heterogeneous platforms (Unity). (1) Alignment Helper Mesh and Alignment Triangle Mesh. (2) Reconstruction of Alignment Helper Meshes and the Alignment Triangle Mesh across Heterogeneous Platforms, with the magenta mesh reconstructed by NVRE; (3) Selection of Target Objects for Alignment; (4) Visualization of Geometry and Dimensions Following Object Alignment.

8. Object-Level Round-Trip Experiment and Evaluation

8.1. Experimental Setup and Objectives

The experiments presented in this section aim to validate whether the proposed dual-structured reconstruction pipeline can preserve consistent geometric structures and functional properties during the semantic-vertex-based primary transformation of IFC objects exported from the Revit platform and their subsequent reconstruction in heterogeneous platforms. The primary objective of this experiment is not merely to confirm the geometric exchangeability of shape data across platforms. Instead, it focuses on verifying whether objects defined based on semantic vertices can continuously preserve the topological consistency of triangular meshes and functional reusability even after platform migration and geometric editing. In other words, this section simultaneously evaluates not only the accuracy of geometric reconstruction but also the maintenance of semantic-driven functional continuity. Since FBX-based visual reconstruction experiments intended solely for visual validation were sufficiently investigated in the fundamental experiments presented in Section 8, they are excluded from this section. Instead, two distinct experimental configurations are designed to comparatively evaluate the structural and functional differences arising from different reconstruction strategies. Both experiments employ the same original IFC isolated footing model and a reinforcing bar (rebar) model additionally created in the Blender environment as common references, while applying different combinations of reconstruction methods to each object.

Specifically, two reconstruction strategies are considered:

- (1) a fully JSON-based reconstruction approach, in which both the isolated footing concrete and the reinforcing bar objects are reconstructed exclusively using JSON representations; and
- (2) a hybrid reconstruction approach, in which the geometrically simpler isolated footing concrete is reconstructed using JSON, whereas the geometrically complex reinforcing bar model is reconstructed using the FBX format, considering its high geometric complexity. The quantitative differences and the preservation of functional properties between these two strategies are systematically compared and analyzed.

Figure 9 illustrates the original objects used in the experiments. The concrete portion of the isolated footing is directly extracted from the IFC file, while the reinforcing bar model is additionally modeled in the Blender environment. This configuration is intended to emulate realistic BIM scenarios that simultaneously include objects with significantly different levels of geometric complexity.

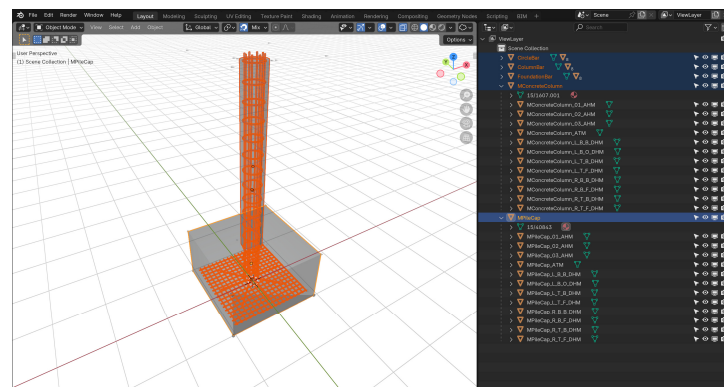


Figure 9. Original experimental BIM objects: concrete footing extracted from IFC and reinforcing bar model (shown in orange) generated in Blender

8.2. 3D Object Reconstruction

This section quantitatively analyzes the geometric reconstruction accuracy, preservation of triangular mesh topology, and data size characteristics of the proposed semantic-vertex-based reconstruction framework through experiments conducted using a pure JSON-based reconstruction strategy.

8.2.1. Pure JSON-Based Reconstruction Strategy (Semantic-Vertex-Based JSON Reconstruction of Both Isolated Footing Concrete and Reinforcing Bar Objects)

This subsection analyzes the results of reconstructing the entire object configuration, including the isolated footing concrete and reinforcing bar objects, solely using semantic-vertex-based JSON data. Under this experimental condition, object reconstruction is performed exclusively based on JSON data, and the scope of analysis is limited to verifying mesh consistency before and after reconstruction and examining data size characteristics.

To validate reconstruction accuracy, a comparative analysis was conducted between the mesh information of the original objects generated in the Blender environment and that of the reconstructed objects in the Unity environment based on identical JSON data. The mesh audit performed in Blender indicated that the experimental dataset contained a total of 8051 vertices and 15,918 triangular faces, with no duplicated vertices or redundant triangles. Using the same evaluation criteria, the mesh statistics of the reconstructed objects in Unity also confirmed that both the vertex and triangle counts were preserved identically to those of the original models.

The exact agreement of mesh statistics between the pre-reconstruction (Blender) and post-reconstruction (Unity) models, at both object-level and total aggregate levels,

demonstrates that semantic-vertex-based JSON reconstruction preserves the topological structure of triangular meshes without loss. Furthermore, no geometric distortions resulting from vertex merging or triangle re-triangulation were observed during the reconstruction process. From a data size perspective, the characteristics of the pure JSON-based reconstruction strategy become clearly evident. For the same experimental dataset, the FBX-based 3D object file size is approximately 482 KB, whereas the pure JSON source file containing both the isolated footing concrete and reinforcing bar objects reaches approximately 58,819 KB (about 58.8 MB), corresponding to a data volume increase of approximately 122 times compared to the FBX format. This substantial difference arises from the text-based nature of JSON, in which dense triangular mesh data and semantic vertex information are explicitly stored without compression. Figure 10 illustrates the visual results of the reconstructed objects in the Unity environment after restoring both the isolated footing concrete and reinforcing bar objects using semantic-vertex-based JSON data.

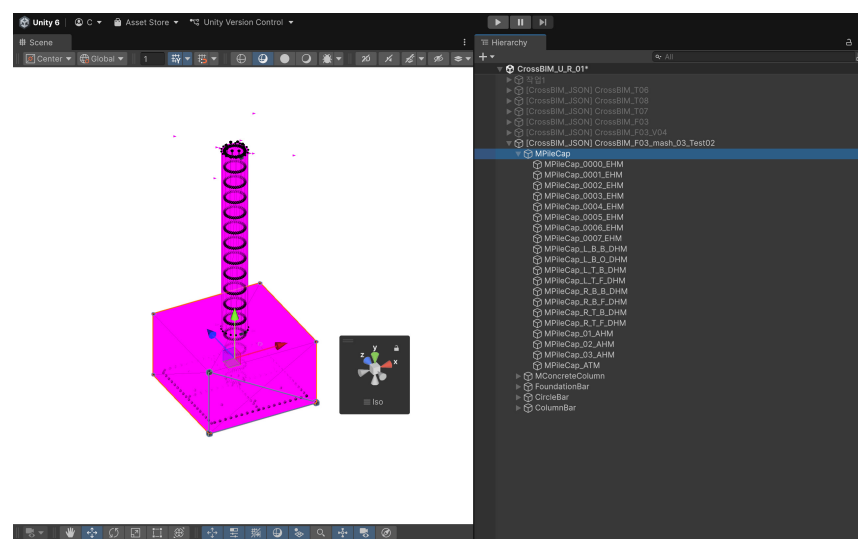


Figure 10. Visual results of pure JSON-based reconstruction of isolated footing concrete and reinforcing bar objects in Unity, including semantic vertices, where the magenta mesh denotes the NVRE-based JSON reconstruction.

8.2.2. Mixed Reconstruction Strategy (JSON + FBX) (JSON-Based Reconstruction of Isolated Footing Concrete and FBX-Based Reconstruction of Reinforcing Bar Objects)

This subsection analyzes the performance of a mixed reconstruction strategy (JSON + FBX), in which the isolated footing concrete structure and the reinforcing bar objects are reconstructed using different data pipelines according to their geometric and functional characteristics. In contrast to the pure JSON-based reconstruction strategy presented in Section 8.2.1, this experimental configuration is designed to reflect practical BIM application scenarios, where heterogeneous objects with significantly different geometric complexity coexist and must be managed efficiently.

Under this experimental condition, the isolated footing concrete structure is reconstructed exclusively using semantic-vertex-based JSON data, whereas the reinforcing bar objects are reconstructed using the conventional FBX-based mesh reconstruction pipeline. Accordingly, the scope of analysis is limited to (1) verifying mesh consistency for the JSON-reconstructed concrete components and (2) evaluating the data size characteristics of the mixed reconstruction configuration, rather than performing full semantic-vertex-based validation of all objects.

To validate reconstruction accuracy, a comparative analysis was conducted between the mesh information of the normalized original concrete structures in the Blender

environment and that of the corresponding models reconstructed in the Unity environment based on identical JSON data. Mesh auditing in Blender indicates that the isolated footing (MPileCap) and concrete column (MConcreteColumn) together consist of 42 vertices and 76 triangular faces, without duplicated vertices or degenerate triangles. Applying the same evaluation criteria in Unity confirms that the reconstructed objects exhibit exactly identical vertex and triangle counts, demonstrating that the topological structure of the concrete components is preserved without loss under the mixed reconstruction strategy.

The exact agreement of mesh statistics between the pre-reconstruction (Blender) and post-reconstruction (Unity) models verifies that semantic-vertex-based JSON reconstruction maintains full geometric and topological integrity even when selectively applied to only part of the object configuration. In contrast, the reinforcing bar objects are reconstructed using the FBX format, whose geometric fidelity is inherently preserved through standard mesh-based data exchange pipelines. Therefore, additional vertex- and face-level equivalence verification for the reinforcement components is not required in this experimental setting.

From a data efficiency perspective, the advantages of the mixed reconstruction strategy become particularly evident. The semantic-vertex-based JSON file representing the concrete components occupies 388 KB, while the FBX file containing the reinforcing bar objects occupies 427 KB, resulting in a total dataset size of approximately 815 KB. Compared to the pure JSON-based strategy in Section 8.2.1, which requires approximately 58.8 MB, the mixed reconstruction approach achieves a data size reduction exceeding two orders of magnitude, while preserving the semantic integrity and geometric fidelity of structurally critical components. This substantial reduction is achieved by separating semantic-rich but geometrically simple concrete elements from geometrically dense but semantically lightweight reinforcement meshes, enabling efficient data management without compromising reconstruction accuracy.

Figure 11 illustrates the reconstructed scene in the Unity environment, where the isolated footing concrete structure is restored using semantic-vertex-based JSON data and the reinforcing bar objects are reconstructed via FBX, demonstrating seamless geometric integration and visual consistency under the proposed mixed reconstruction strategy.

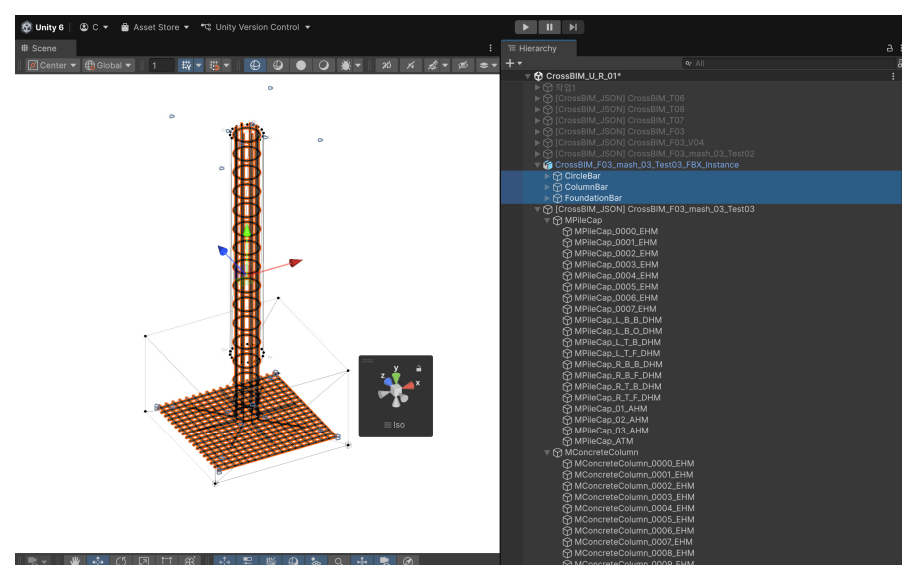


Figure 11. Reconstructed scene in Unity using the mixed reconstruction strategy, comprising JSON-based restoration of isolated footing concrete and FBX-based reconstruction of reinforcing bar objects, in which the reinforcing bars are rendered in orange.

8.3. Post-Reconstruction Round-Trip Validation

This section evaluates the robustness and stability of the proposed reconstruction framework through round-trip validation experiments, in which reconstructed objects are edited, re-saved, and subsequently restored back to the original modeling environment. The objective is to verify the preservation of geometric identity, semantic consistency, and functional continuity throughout the complete reconstruction–editing–reconstruction cycle.

8.3.1. Round-Trip Validation of the Pure JSON-Based Reconstruction Strategy

Figure 12 illustrates the round-trip reconstruction process and its results after restoring both the isolated footing concrete and reinforcing bar objects using semantic-vertex-based JSON data. In this experiment, the reconstructed objects are edited and re-saved in the Unity environment and subsequently restored back to the Blender environment, forming a complete round-trip reconstruction pipeline. The objective of this experiment is to verify the preservation of geometric identity and the continuity of semantic-based functional structures throughout the round-trip process.

Figure 12(a) presents the final result restored in Blender after round-trip reconstruction, following editing and re-saving in Unity. In this process, all semantic-vertex-based auxiliary meshes—including the Dimension Helper Mesh (DHM), Edit Helper Mesh (EHM), Alignment Helper Mesh (AHM), and Alignment Triangle Mesh (ATM)—are correctly reconstructed. In particular, although the edit helper mesh is initially defined as a virtual reference structure within the JSON data, it is restored as a visually explicit mesh object after the round-trip, maintaining an identical structural state in both Unity and Blender environments.

Figure 12(b) shows the original object restored in Unity from semantic-vertex-based JSON data, while Figure 12(c) presents the results of geometric editing of the concrete shape and reinforcement length using the edit helper mesh. Based on the modified geometry, the dimension measurement function is reapplied through the dimension helper mesh, confirming that semantic-vertex-based functional operations are continuously preserved even after editing.

Figure 12(d) visualizes the final Blender-restored model overlaid with the original pre-reconstruction model in a projection-aligned mesh representation, enabling intuitive verification of geometric equivalence and structural consistency before and after the round-trip process. This result clearly demonstrates that the proposed framework ensures lossless geometric restoration and stable functional continuity throughout the complete round-trip reconstruction cycle.

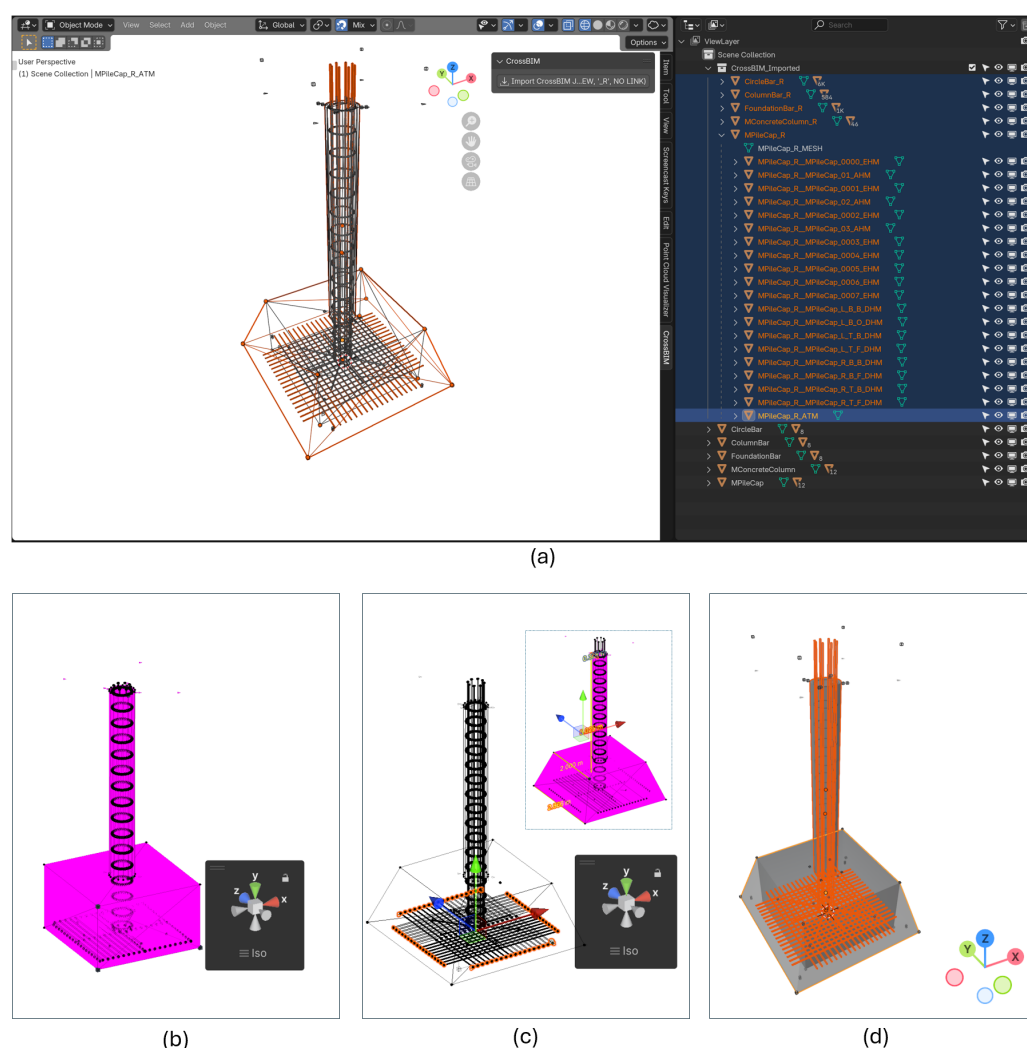


Figure 12. Round-trip reconstruction results using the pure JSON-based strategy. (a) Final Blender restoration after round-trip reconstruction; (b) Original model restored in Unity from semantic-vertex-based JSON, where the pink mesh represents the NVRE-based JSON reconstruction; (c) Geometric editing using the Edit Helper Mesh and re-application of dimensions, with the pink mesh indicating the JSON-reconstructed geometry; (d) Overlay comparison of pre- and post-reconstruction models in Blender, in which the orange color denotes the reinforcing bar model.

8.3.2. Round-Trip Validation of the Mixed Reconstruction Strategy (JSON + FBX) (Semantic-Vertex-Based JSON Reconstruction of Isolated Footing Concrete and FBX-Based Reconstruction of Reinforcing Bar Objects)

In this experiment, a round-trip reconstruction process was performed on a mixed object configuration consisting of an isolated footing concrete object reconstructed using semantic-vertex-based JSON and reinforcing bar objects reconstructed using the FBX format, in order to analyze the post-round-trip geometric stability and structural consistency.

During this process, the FBX-based reinforcing bar model was repositioned onto the isolated footing concrete object through scaling and translation, while the concrete component, after JSON-based reconstruction, was geometrically edited using the edit helper mesh (Edit Helper Mesh, EHM). Subsequently, based on the modified configuration, the entire object assembly was restored back to the Blender environment to validate the round-trip reconstruction results.

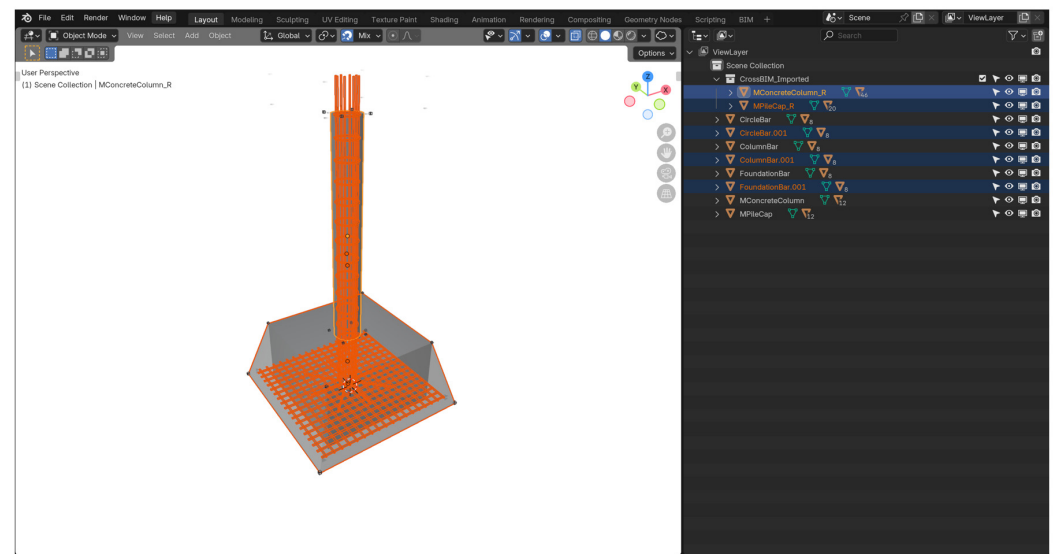
Figure 13(a) presents the final result restored in Blender after editing and re-saving in the Unity environment. In this process, the isolated footing concrete object reconstructed using JSON correctly preserves all semantic-vertex-based auxiliary mesh

structures, including the Dimension Helper Mesh (DHM), Edit Helper Mesh (EHM), Alignment Helper Mesh (AHM), and Alignment Triangle Mesh (ATM). In contrast, the reinforcing bar objects reconstructed using the FBX format remain in their initial restored state, without generating semantic-vertex-based auxiliary mesh structures.

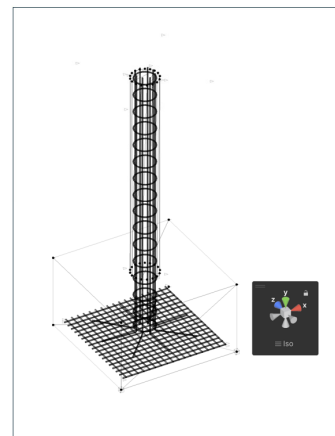
Figure 13(b) shows the initial mixed reconstruction state in the Unity environment, while Figure 13(c) illustrates the results after repositioning the FBX-based reinforcing bar model through scaling and translation, followed by geometric editing of the concrete object using the edit helper mesh and the subsequent reapplication of dimension measurements. These results confirm that, even under the mixed reconstruction strategy, the semantic-vertex-based functional structure of JSON-reconstructed objects is robustly preserved after geometric editing.

Figure 13(d) visualizes the final Blender-restored model overlaid with the original pre-reconstruction model in a projection-aligned mesh representation, demonstrating that the overall geometric configuration and spatial alignment are visually consistent before and after the round-trip process, even under the mixed reconstruction strategy.

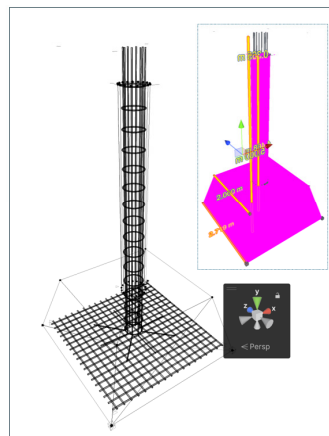
In summary, the experimental results confirm that the overall object configuration is stably preserved after round-trip reconstruction using the mixed strategy (JSON + FBX). However, due to the characteristics of FBX-based reconstruction, vertex-level geometric editing and rigorous shape equivalence verification are not supported for the reinforcing bar objects. As a result, although visual consistency is maintained, strict geometric identity at the structural level cannot be guaranteed. This indicates that the mixed reconstruction strategy prioritizes visual stability and practical efficiency, whereas the pure JSON-based reconstruction strategy is more suitable for applications requiring precise semantic-based functional reuse and vertex-level geometric consistency.



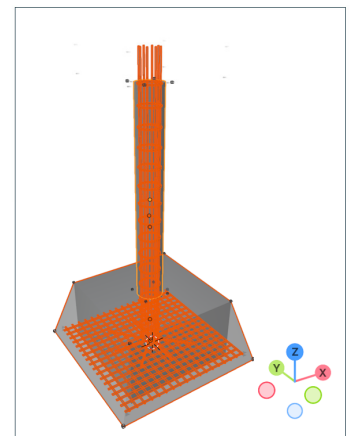
(a)



(b)



(c)



(d)

Figure 13. Round-trip reconstruction results using the mixed strategy (JSON + FBX). (a) Final Blender restoration after mixed round-trip reconstruction, where the orange elements correspond to FBX-based reinforcing bar reconstruction; (b) Initial mixed reconstruction state in Unity (JSON concrete + FBX reinforcement); (c) Repositioning of FBX reinforcement and geometric editing of JSON concrete with re-applied dimensions, where the magenta mesh denotes the JSON-reconstructed concrete geometry; (d) Overlay comparison of pre- and post-reconstruction models in Blender, in which the orange color represents the FBX-reconstructed reinforcing bar model.

8.4. Comparative Evaluation and Interpretation of Reconstruction and Round-Trip Performance

This section provides an integrated comparative evaluation of the reconstruction and round-trip experiments presented in Sections 8.2.1–8.3.2, together with a concise geometric interpretation of triangle mesh identity preservation. The comparison focuses on data size, reconstruction strategy, vertex and triangle count identity, editing capability, round-trip stability, and practical efficiency, which collectively characterize the trade-off between geometric precision and computational efficiency.

Table 1 summarizes the quantitative performance of pure JSON-based and mixed JSON + FBX reconstruction strategies, including both initial reconstruction and round-trip validation results. In particular, vertex and triangle count identity is adopted as the primary quantitative indicator for evaluating geometric consistency and topological preservation.

Table 1. Quantitative comparison of reconstruction and round-trip strategies.

Metric	Pure JSON Reconstruction (8.2.1)	Mixed Reconstruction (8.2.2)	Pure JSON Round-Trip (8.3.1)	Mixed Round-Trip (8.3.2)
Total data size	58.8 MB	0.8 MB	58.8 MB	0.8 MB
Reconstruction	JSON	JSON + FBX	JSON	JSON + FBX
Semantic coverage	Full	Partial	Full	Partial
Vertex count identity	(8051 = 8051)	(42 = 42) *	–	*
Triangle identity	(15,918 = 15,918)	(76 = 76) *	–	*
Editing capability	Vertex-level	Partial	Vertex-level	Partial
Round-trip fidelity	–	–	Structural identity	Visual consistency
Practical efficiency	Low	High	Low	High

* Vertex- and triangle-level identity verification applies only to JSON-reconstructed concrete components. FBX-based reinforcement objects were excluded from this verification.

The quantitative results indicate that the pure JSON-based strategy achieves complete vertex- and triangle-level identity, full semantic continuity, and exact round-trip reconstruction, thereby enabling lossless geometric recovery and precise functional reuse. However, these advantages are accompanied by substantially increased data volume and computational overhead, which may limit scalability for large-scale BIM datasets.

In contrast, the mixed JSON + FBX strategy significantly reduces storage requirements and computational cost, while maintaining visual consistency and stable spatial configuration throughout both reconstruction and round-trip processes. Although vertex-level identity verification is not applicable to FBX-based reinforcement components, the mixed strategy offers a practical compromise between geometric reliability and computational efficiency, making it particularly suitable for visualization, collaborative coordination, and real-time BIM applications.

The geometric principle underlying triangle mesh identity preservation in the proposed framework can be summarized as follows. A triangular mesh M is represented by a vertex coordinate set V and a triangle connectivity set T :

$$M = (V, T). \quad (1)$$

Each Edit Helper Mesh (EHM) is defined as a semantic vertex control element that maintains a fixed one-to-one correspondence with a triangle mesh vertex. During shape editing, updated semantic vertex positions generate a new vertex coordinate set V' , while the original connectivity T remains unchanged. Consequently, a regenerated mesh is constructed as

$$M' = (V', T). \quad (2)$$

This regeneration-based mechanism ensures that geometric modification is achieved through full mesh reconstruction rather than incremental deformation, thereby eliminating cumulative numerical errors, preventing topological distortion, and ensuring stable round-trip reconstruction performance.

Overall, the experimental and theoretical analyses confirm that the pure JSON-based reconstruction strategy is best suited for precision-critical BIM applications, such as parametric modeling, automated dimensioning, semantic-driven analysis, and digital twin systems, where exact geometric identity and functional continuity are essential. Conversely, the mixed JSON + FBX strategy is more appropriate for large-scale visualization, immersive environments, and collaborative workflows, where lightweight data handling, real-time performance, and practical scalability are prioritized. The proposed dual-structured framework therefore provides a flexible and robust solution for balancing precision and efficiency across diverse BIM application scenarios.

9. Project-Level Reconstruction Using an IFC Project

This section presents a project-level reconstruction experiment conducted on a complete BIM project exported from Autodesk Revit in IFC format. The objective of this experiment is to validate the scalability, geometric stability, and structural consistency of the proposed framework at the full-project scale. All IFC objects were preprocessed in the Blender environment through geometry extraction, triangulated mesh normalization, coordinate normalization, and vertex index stabilization. Subsequently, the reconstructed data were imported into the Unity environment using dual parallel pipelines consisting of semantic-vertex-based JSON reconstruction (NVRE) and FBX-based visual reconstruction (VRE).

In the full-project experiment, the total storage size of the JSON-based non-visual reconstruction entity (NVRE) was approximately 200 MB, whereas the same model converted into a visual reconstruction entity (VRE) using the FBX format resulted in a file size of approximately 4 MB. This substantial difference reflects the fact that NVRE preserves not only visual geometric information but also semantic vertices, topological structures, object relationships, and functional attributes, whereas VRE mainly stores rendering-oriented geometric representations.

Figure 14 illustrates the experimental workflow and results of the project-level triangulated mesh normalization and dual-parallel reconstruction pipeline. Figure 14(1) shows the Blender-based preprocessing pipeline applied to the entire IFC project, including geometry extraction, triangulated mesh normalization, coordinate normalization, and vertex index stabilization. At this stage, all IFC objects, regardless of their geometric complexity or IFC classification, are converted into normalized triangulated meshes, eliminating duplicated vertices, overlapping surfaces, micro-gap faces, non-manifold configurations, and numerically unstable geometric components, thereby establishing a geometrically consistent and numerically robust representation basis for subsequent reconstruction.

As illustrated in Figure 14(2), the normalized triangulated mesh data are then imported into Unity through parallel VRE and NVRE pipelines to reconstruct the entire project scene. The results confirm that the majority of objects are stably reconstructed with accurate geometric configurations and consistent spatial alignment, demonstrating that project-level reconstruction can be achieved without cumulative geometric distortion, coordinate drift, or structural degradation. However, in a limited number of cases, localized visual or topological instabilities were observed during direct JSON-based reconstruction. Detailed analysis indicates that these artifacts are not attributable to geometric complexity, IFC entity types, or topological deficiencies, but rather originate from numerical instability occurring at floating-point precision boundary conditions within the GPU rendering pipeline. Even when geometric shapes and vertex coordinates remain identical, floating-point boundary sensitivity in GPU computations may lead to transient rendering instabilities in a small subset of triangles.

To resolve this issue, a global remeshing-based re-triangulation strategy was applied prior to JSON generation, as shown in Figure 14(3). In this remeshing process, the original triangle mesh is completely discarded and regenerated, while the original vertex positions are strictly preserved. This procedure does not modify the underlying geometry or topology but stabilizes floating-point boundary conditions in GPU computation by regenerating the triangulated mesh while preserving original vertex positions, thereby eliminating the observed instabilities. Experimental results confirm that, after applying this global remeshing process, all visual artifacts were completely removed, and the instability did not reoccur even without further remeshing, indicating that the phenomenon arises from transient numerical instability rather than persistent geometric defects.

Notably, stable reconstruction was also achieved for highly complex and non-standard IFC geometries, including IfcSweptDiskSolid-based curved beams, IfcBSplineSurface-based freeform envelopes, composite walls with irregular openings, and complex assemblies integrating structural, reinforcement, and mechanical components. These results demonstrate that triangle-mesh-based normalization provides a universal and robust foundation for project-scale cross-platform BIM reconstruction.

Overall, the experimental results demonstrate that the proposed framework effectively scales from isolated object reconstruction to full-scale BIM project restoration, ensuring geometric consistency, spatial coherence, and functional continuity, thereby supporting practical applications in digital twin environments, immersive visualization, collaborative BIM workflows, and sustainable BIM data management.

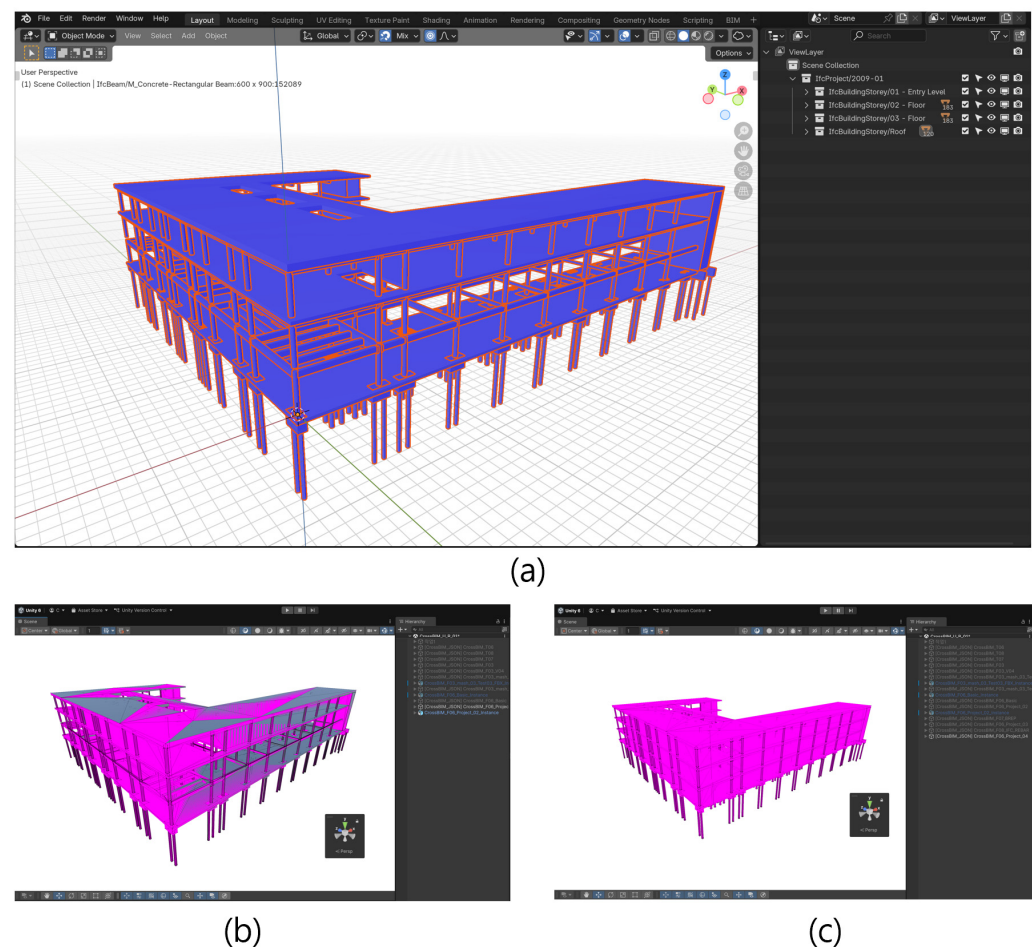


Figure 14. Overall results of triangle mesh normalization and the dual-parallel reconstruction pipeline at the IFC project level. (a) All Revit-exported IFC objects normalized into triangulated meshes in the Blender environment, where the blue surfaces indicate the original face normal directions; (b) Overlaid visualization in Unity of objects reconstructed through parallel VRE and NVRE pipelines, showing consistent geometric alignment after project-level reconstruction (gray: FBX reconstruction; magenta: JSON reconstruction); (c) Pure JSON-based reconstruction of the complete IFC project in Unity, including previously failed IFC objects, using a global remeshing strategy based on triangle mesh normalization and regenerated triangulation (magenta mesh: JSON reconstruction).

The quantitative results presented in Tables 2 and 3 provide strong experimental evidence that the complete pipeline adopted in this study—comprising Blender-based geometric preprocessing and triangulated mesh normalization followed by Unity-based parallel FBX and JSON reconstruction—operates reliably at the project scale in terms of

geometric accuracy, structural consistency, and numerical robustness. In particular, the results demonstrate that geometric integrity and topological consistency can be maintained throughout the entire cross-platform workflow, even when processing a large-scale BIM project consisting of hundreds of heterogeneous IFC objects.

According to the results in Table 2, most geometric quality indicators, including duplicate vertices, duplicate triangles, degenerate triangles, tiny-area triangles, line-like triangles, short-edge triangles, and flipped vertex normals, remain zero even in the original imported state, confirming that no significant geometric degradation occurs during the IFC-to-Blender conversion process. In contrast, a substantial proportion of flipped triangles (37.3%) is observed in the original state, primarily attributable to differences in triangulation strategies and face normal interpretation during IFC import. However, after applying the proposed triangulated mesh normalization and orientation-consistency procedures, the proportion of flipped triangles is completely eliminated (0%), and this result remains unchanged after remeshing. This demonstrates that the proposed preprocessing pipeline fundamentally stabilizes mesh orientation and topological consistency without inducing any geometric loss, providing a robust geometric foundation for subsequent reconstruction.

The results in Table 3 further confirm that the triangulated meshes normalized in Blender are accurately reconstructed in the Unity environment through both FBX-based and JSON-based parallel pipelines while preserving identical geometric structures and spatial coherence. Both reconstruction approaches exhibit exactly matching object counts (850), total triangle counts (55,202), unique local vertex counts (29,285), and unique world vertex counts (26,808), thereby confirming perfect structural equivalence at both the object and scene scales. This indicates that JSON-based reconstruction achieves geometric fidelity fully comparable to that of conventional FBX-based visual reconstruction, ensuring that no cumulative geometric drift or scene-level distortion is introduced during cross-platform transfer.

In contrast, a pronounced difference emerges in terms of raw vertex counts. The FBX pipeline exhibits a 2.86-fold increase in raw vertices (83,660) due to rendering-driven vertex splitting, whereas the JSON pipeline preserves the original minimal geometric representation without any vertex splitting (29,285), yielding a Raw/Unique ratio of exactly 1.00 in the JSON pipeline. This confirms zero rendering-driven vertex splitting and strict preservation of the original geometric vertex representation, quantitatively demonstrating that JSON-based reconstruction offers superior structural efficiency and data compactness while maintaining strict geometric fidelity. From a practical perspective, this structural compactness contributes directly to reduced memory consumption, faster data transfer, and improved scalability for large-scale BIM visualization and analysis tasks.

Furthermore, both pipelines preserve identical distributions of outward- and inward-facing triangles, and the small proportion of inward-facing faces (1.90%) is interpreted not as geometric or topological error, but as numerical instability arising from floating-point boundary conditions in ultra-thin geometric regions during GPU-based rendering. This quantitatively supports the conclusion that the localized visual artifacts observed in JSON-based reconstruction originate from numerical rendering limitations rather than from geometric degradation, reinforcing the robustness of the proposed geometric normalization strategy.

Overall, the results in Tables 2 and 3 quantitatively demonstrate that the proposed triangulated-mesh-based normalization combined with semantic-vertex-based JSON reconstruction provides superior geometric accuracy, structural consistency, numerical robustness, and data efficiency compared to conventional FBX-based reconstruction pipelines, and can be reliably applied to project-scale BIM reconstruction.

Table 2. Quantitative evaluation of geometric preprocessing in Blender prior to reconstruction.

Metric	Definition	Original State (Blender Im- port)	Triangulated + Parametric Mesh	Remeshed
Raw Vertices	Total number of mesh vertices (vertex buffer size)	29,285	29,285	29,285
Total Triangles	Total number of triangular faces	55,202	55,202	55,202
Duplicate Local Vertices	Vertices sharing identical local spatial coordinates	0	0	0
Duplicate Triangles	Triangles sharing identical vertex triplets	0	0	0
Flipped Triangles	Triangles whose geometric normal direction is inconsistent with the average vertex normal orientation ($\text{dot}(\text{triN}, \text{avgVertN}) < 0$)	20,604 (37.325%)	0 (0.000%)	0 (0.000%)
Degenerate Triangles	Triangles with near-zero area ($\text{area} < 1 \times 10^{18}$)	0	0	0
Tiny-Area Triangles	Triangles with extremely small area relative to bounding-box scale ($\text{area} < 1 \times 10^{10} \times \text{diag}^2$)	0	0	0
Line-Like Triangles	Nearly collinear triangles with extremely low shape quality ($q \leq 0.001$)	0	0	0
Short-Edge Triangles	Triangles containing extremely short edges ($\text{min_edge} < 1 \times 10^6 \times \text{diag}$)	0	0	0
Flipped Normals	Vertex normals explicitly oriented opposite to surface normals	0	0	0
Bad Triangles	Aggregated count of degenerate, tiny-area, line-like, or short-edge triangles	0	0	0

Table 3. Structural comparison of FBX and JSON-based reconstruction pipelines in Unity.

Metric	FBX Reconstruction	JSON Reconstruction	Interpretation
Objects Counted	850	850	Identical object set ensures fair comparison
Total Triangles	55,202	55,202	Identical topology and geometric fidelity
Unique Local Vertices	29,285	29,285	Identical geometric vertex sets → perfect shape equivalence
Unique World Vertice	26,808	26,808	Identical scene-scale spatial structure
Raw Vertices	83,660	29,285	FBX expands the vertex buffer due to rendering-driven vertex splitting, whereas JSON preserves a strictly minimal geometric representation
Raw/Unique Ratio	2.86×	1.00×	Quantifies rendering-driven vertex expansion in FBX and confirms zero vertex splitting in JSON
Outward Faces	54,155	54,155	Identical face orientation
Inward Faces	1047 (1.8967%)	1047 (1.8967%)	Numerical instability in ultra-thin regions, not geometric error
Degenerate/Tiny/Line-like	0/0/0	0/0/0	High mesh quality preserved in both pipelines

10. Discussion

This section provides a comprehensive interpretation of the experimental results presented in Sections 8 and 9 and systematically discusses the technical significance, reconstruction safety and structural implications, academic contributions, practical implications, and future research directions of the proposed invariant triangle mesh-based non-

visual reconstruction framework. In particular, the logical meaning and technical implications of the proposed approach are analyzed from the perspectives of object identity preservation, functional continuity, and cross-platform scalability.

Furthermore, based on the project-scale reconstruction experiments, this section integratively discusses the remeshing strategy for graphical stabilization, the necessity of geometric preprocessing prior to reconstruction, and the comparative performance of FBX-based visual pipelines and JSON-based non-visual reconstruction pipelines, thereby clarifying that the proposed framework extends beyond simple mesh transformation and establishes a platform-independent logical reconstruction paradigm. In addition, the potential of the JSON-based semantic-vertex data structure to serve as training data for AI-based algorithms for vertex exploration and semantic vertex inference is discussed, highlighting the long-term academic and technological extensibility of the proposed framework.

10.1. Integrated Interpretation of Reconstruction Results and Technical Significance

The experimental results demonstrate that the proposed invariant triangle mesh-based non-visual reconstruction pipeline can consistently preserve geometric identity and semantic functional reference structures after cross-platform transformations. In the object-level experiments conducted on an IFC isolated footing, the parent object mesh was consistently regenerated by each platform's native geometric engine using the invariant triangle mesh stored in the non-visual reconstruction entity as the reconstruction reference. Simultaneously, semantic-vertex-based helper meshes for dimensioning, editing, and alignment were restored while maintaining identical logical reference relationships.

The object-level round-trip transformation experiments further confirmed that vertex indices, triangle topological structures, and semantic reference criteria remain stable even after multiple platform transitions and intermediate modifications. These results indicate that the proposed framework goes beyond simple mesh regeneration and instead establishes a logical reconstruction mechanism capable of simultaneously preserving platform-independent object identity and functional continuity.

Conventional BIM interoperability and object reconstruction studies primarily rely on platform-specific tessellation strategies, geometric kernels, and mesh optimization algorithms, which inevitably cause variations in vertex configurations, triangle connectivity, and face orientations during platform transitions. Although visual similarity may be largely preserved, such approaches cannot structurally guarantee the continuity of semantic-vertex-based functions or user-defined reference systems.

In contrast, this study explicitly defines invariant triangle meshes and vertex ID-based topological structures as platform-independent reconstruction criteria and logically integrates semantic vertices and helper meshes into this invariant reference framework. As a result, object identity preservation is achieved not through post hoc geometric comparison, but through predefined logical reconstruction principles. Accordingly, platform-native geometric engines are repositioned as execution layers rather than authoritative reconstruction criteria, and platform-dependent geometric inconsistencies are structurally mitigated.

Consequently, the proposed approach fundamentally transforms BIM object reconstruction from a geometric reconfiguration problem into a logical re-identification problem, providing a new theoretical foundation for robust object identity preservation and functional continuity across heterogeneous BIM and 3D platforms.

10.2. Stability and Structural Implications of Project-Level Reconstruction

While object-level experiments validated the logical soundness of the proposed reconstruction framework, project-scale experiments served to verify its scalability, numerical robustness, and execution stability. Large-scale BIM projects inherently contain various geometric irregularities—including degenerate triangles, ultra-thin gaps, duplicated vertices, unstable face orientation judgments, and sliver triangles—caused by differences in modeling kernels, tessellation strategies, and floating-point computation environments across platforms. These irregularities often result in missing faces, back-face culling errors, flickering, and distorted normals in real-time visualization and game-engine-based environments, which can directly compromise object identity and functional reference continuity.

To address these issues, this study integratively applied remeshing-based graphical stabilization, geometric preprocessing prior to reconstruction, and a JSON-based non-visual reconstruction pipeline. The experimental results showed that controlled preprocessing and remeshing can effectively suppress numerical instabilities—such as floating-point noise, face orientation ambiguity, and vertex duplication—without altering the total triangle count or geometric fidelity, thereby achieving consistent execution stability across platforms. This confirms that remeshing in this framework functions not as a geometric simplification technique but as a numerical stabilization layer enabling platform-independent logical reconstruction.

Moreover, the comparative analysis between FBX-based visual pipelines and JSON-based non-visual reconstruction pipelines clearly demonstrated that preserving invariant vertex indices and triangle topological structures is far more critical than visual similarity for maintaining object identity and functional continuity. In FBX-based workflows, implicit re-triangulation and vertex reordering frequently occur during import and export operations, leading to partial degradation of semantic reference consistency. In contrast, the JSON-based pipeline explicitly preserves vertex indices, triangle connectivity, and semantic reference relationships, thereby ensuring complete logical consistency even after multiple round-trip transformations.

These findings experimentally validate that the proposed non-visual reconstruction entity-based pipeline fundamentally extends conventional visual interoperability approaches and enables stable, platform-independent logical reconstruction at the project scale.

10.3. Academic Contributions

The primary academic contribution of this study lies in shifting the paradigm of BIM object reconstruction from visual similarity-oriented approaches toward logically defined object identity preservation. By introducing the invariant triangle mesh as a platform-independent reconstruction reference and structuring semantic vertices and helper meshes upon this foundation, the proposed framework clearly separates platform-dependent geometric realization from platform-independent reconstruction logic.

Whereas previous semantic-vertex-based studies have mainly focused on single-purpose applications, such as automatic dimension generation or topology-based performance analysis, this study redefines semantic vertices as identity anchors rather than function-specific references. By ensuring the invariance of semantic vertices across platform transformations, the framework enables the consistent reuse of multiple semantic-driven functions—including dimensioning, editing, and alignment—on a single reconstructed object identity. This transition from function-centered semantic utilization to identity-centered multi-functional reuse constitutes the core academic advancement of this work.

Furthermore, by integrating remeshing-based numerical stabilization and preprocessing-based conditioning strategies, this study extends semantic reconstruction research toward large-scale, real-time, and immersive execution environments, thereby bridging theoretical reconstruction logic and industrial-scale computational execution.

10.4. Practical Implications and Cross-Platform Scalability

This study provides meaningful practical implications for both BIM managers and software developers operating in heterogeneous BIM and 3D environments.

From a BIM management perspective, interoperability should no longer be evaluated solely based on visual geometric similarity but rather on the preservation of platform-independent logical reconstruction criteria. For long-term data preservation and sustainable digital asset reuse, workflows that maintain invariant reconstruction references—such as semantic vertices defined on invariant triangle meshes—represent effective and sustainable BIM management strategies.

From a software development perspective, instead of repeatedly redefining triangulation and geometric reconstruction rules for individual platforms, it is more desirable to adopt semantic-vertex-based triangulated mesh reconstruction as a common logical foundation and focus development efforts on high-level functional modules, including dimensioning, editing, alignment, quantity takeoff, schedule management, material property handling, structural analysis, and automated modeling. This paradigm enhances functional consistency across platforms while significantly improving development efficiency.

The reconstruction and round-trip transformation pipeline adopted in this study can be generalized as follows:

Native BIM authoring environments $\rightleftharpoons$ IFC $\rightleftharpoons$ triangulated mesh normalization + parallel JSON pipeline $\rightleftharpoons$ heterogeneous BIM, visualization, and immersive platforms.

This pipeline can be concretely instantiated, for example, as:

Revit/Archicad $\rightleftharpoons$ IFC $\rightleftharpoons$ Tri-mesh normalization + JSON $\rightleftharpoons$ Revit/Archicad/Unity/Unreal Engine.

Within this framework, object-level logical reconstruction criteria were experimentally extended to project-level workflows, while inter-object dependencies—such as beam–column connections, global coordinate hierarchies, and spatial placement relationships—were preserved using IFC-defined relationships. This demonstrates that stable project-scale reconstruction can be achieved without introducing additional project-specific reconstruction logic, thereby reducing redundant modeling, minimizing digital waste, and extending the effective lifecycle of BIM assets in support of sustainable cross-platform BIM ecosystems.

10.5. Limitations and Future Research Directions

Although this study experimentally validated the feasibility of invariant triangle mesh-based logical reconstruction at both object and project levels, further research is required to investigate optimal instantiation strategies within diverse platform environments. Future studies should systematically explore adaptive reconstruction architectures that account for differences in geometric kernels, modeling paradigms, and parametric object systems while preserving platform-independent reconstruction criteria.

In addition, further investigation into automatic numerical instability detection, dynamic tolerance control, and platform-aware adaptive remeshing strategies will be essential for enhancing execution robustness in real-time digital twin platforms, immersive XR environments, and ultra-large infrastructure models.

Notably, the JSON-based semantic vertex data structure established in this study functions as clean data in which geometric and topological information are normalized and numerical noise is suppressed, providing important potential for use as training data

for AI-based algorithms for vertex exploration and semantic vertex inference. Such data structures can support automated semantic vertex detection, improved accuracy, and enhanced generalization, thereby laying a foundation for future extensions toward AI-based dimension automation, intelligent object interpretation, and automated digital twin alignment technologies.

11. Conclusions

This study addressed two fundamental research questions in cross-platform BIM reconstruction: (RQ1) how the geometric identity of BIM objects can be preserved using platform-independent criteria during cross-platform transformation and reconstruction, and (RQ2) how semantic-based functions can be restored and reused after geometric reconstruction. To this end, this study proposed an integrated framework that combines an invariant triangle mesh-based non-visual reconstruction pipeline with a semantic-vertex-based functional restoration mechanism, thereby providing a unified solution to both challenges.

To address RQ1, this study defined the invariant triangle mesh as a platform-independent criterion for geometric and topological reconstruction and established a logical reconstruction architecture in which platform-native geometric engines operate as execution layers. The experimental results verified that this approach consistently preserves the geometric identity and topological structure of BIM objects after cross-platform transformation. In particular, the IFC project-scale experiments demonstrated that remeshing-based preprocessing effectively eliminates numerical and geometric instabilities, including micro-gaps, duplicated vertices, ultra-thin triangles, and unstable face-orientation judgments, thereby significantly reducing missing faces and rendering artifacts after platform transitions.

To address RQ2, semantic vertices were redefined as stable anchors of object identity, and a consistent mechanism was established for restoring and reusing semantic-based functions, including dimension generation, geometric editing, and object alignment. Unlike conventional approaches that redefine functionality after geometric reconstruction, the proposed framework adopts a logic-centered reconstruction paradigm, in which functional criteria are restored based on preserved geometric identity, thereby structurally ensuring the continuity and reusability of semantic-based functions.

At the object level, the experiments confirmed that both geometric structures and semantic-based functions remain stable after platform round-trip transformations and intermediate modifications and that full equivalence is preserved upon returning to the original platform. At the project level, the dual-parallel reconstruction pipeline, combining JSON-based non-visual reconstruction and FBX-based visual reconstruction, was shown to provide robust overall reconstruction performance. In particular, the project-scale comparative experiments demonstrated that the JSON-based non-visual reconstruction pipeline maintains complete structural consistency, with no changes in vertex count, vertex indices, or triangle topological structures throughout the reconstruction process. This stands in clear contrast to FBX-based visual reconstruction, in which partial vertex reordering and localized re-triangulation were observed, and experimentally verifies that the JSON-based reconstruction structure provides superior robustness for preserving object identity and logical reference relationships.

The primary limitation of this study lies in the restriction of experimental validation to the Blender–Unity round-trip transformation environment. Therefore, further cross-platform experiments involving diverse commercial BIM authoring tools and real-time immersive execution platforms are required to comprehensively evaluate the generality, scalability, and industrial applicability of the proposed framework.

Overall, this study simultaneously resolves two core challenges of cross-platform BIM reconstruction—object identity preservation and semantic-based functional continuity and reusability—by integrating platform-independent geometric reconstruction criteria with semantic-vertex-based functional restoration mechanisms into a unified framework. From an academic perspective, this study establishes a formal theoretical foundation for a paradigm shift from visually oriented reproduction to logic-centered reconstruction. From an industrial perspective, it provides a practical technological basis for long-term BIM asset reuse, digital waste reduction, and the development of sustainable digital construction environments.

Finally, the standardization of semantic-vertex-based invariant triangle mesh representations is identified as a critical future research challenge for ensuring object identity preservation and functional continuity across BIM, digital twin, and XR-based construction platform ecosystems, highlighting the necessity of sustained academic and industrial collaboration toward this goal.

12. Patents

A patent application related to the proposed framework has been filed. To ensure clarity regarding its availability, the following principles are defined. Use of the framework for academic and non-commercial research purposes is fully permitted and encouraged to support reproducibility and scholarly validation. For commercial applications, industrial deployment, or large-scale distribution, appropriate consultation and agreement are required in accordance with the scope of the filed patent and applicable licensing terms. Selected implementation components and industrial licenses may be provided upon mutual agreement, with the aim of balancing open academic use and the responsible protection of intellectual property. These policies are intended to promote academic dissemination and collaborative research, while ensuring appropriate protection and sustainable utilization of the proposed technology in industrial environments.

Author Contributions: The author conceived and designed the study, developed the models and methodology, conducted all experiments and analyses, and wrote the entire manuscript.

Funding: The present research was supported by the research fund of Dankook University in 2024.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All geometric models, semantic vertex definitions, and experimental datasets were generated and modeled by the author. The IFC objects used in the experiments were obtained from publicly available open IFC repositories. As a patent application related to this research is currently under review, the processed datasets and derived data are not publicly available; however, they may be provided by the author upon reasonable request.

Conflicts of Interest: The author declares no conflicts of interest related to this study. The funding body had no role in the design of the study; in the collection, analysis, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

AHM	Alignment Helper Mesh
ATM	Alignment Triangle Mesh (reference triangle structure for alignment)
BIM	Building Information Modeling
DHM	Dimension Helper Mesh

EHM	Edit Helper Mesh
FBX	Filmbox format (Autodesk interchange format)
HDF5	Hierarchical Data Format version 5
IFC	Industry Foundation Classes
JSON	JavaScript Object Notation
JSON-BIN	JavaScript Object Notation with Binary Payload
MR	Mixed Reality
AR	Augmented Reality
VR	Virtual Reality
SV	Semantic Vertex
UVI	Unique Vertex Index

Appendix A. Original JSON Data for Non-Visual Object Representation

This appendix provides the original JSON data used in this study for non-visual object reconstruction and round-trip validation.

The data corresponds to a basic cube object exported from the Blender platform and is included verbatim to demonstrate the proposed JSON-based representation. No modification or reduction has been applied to the data shown below.

```
{
  "scene_meta": {
    "scene_name": "Scene",
    "blend_file": "CrossBIM_F04",
    "schema_version": "3.3.6-extended-lite-roundtrip-locked-names",
    "timestamp_utc": "2025-12-28T13:46:02Z",
    "export": {
      "fbx_path": "C:\\Users\\cjh\\Desktop\\P_P_01\\CrossBIM_F04.fbx",
      "json_path": "C:\\Users\\cjh\\Desktop\\P_P_01\\CrossBIM_F04.json"
    },
    "scene_object_origins": [
      {
        "name": "Cube",
        "name_raw": "Cube",
        "type": "MESH",
        "kind": "other",
        "origin_world": [0.0, 0.0, 0.0]
      }
    ]
  },
  "objects": [
    {
      "object": {
        "name": "Cube",
        "data_name": "Cube.015",
        "transform": {
          "location_world": [0.0, 0.0, 0.0],
          "rotation_euler_world": [0.0, -0.0, 0.0],
          "scale_world": [1.0, 1.0, 1.0],
          "matrix_world": [
            [1.0, 0.0, 0.0, 0.0],
            [0.0, 1.0, 0.0, 0.0],
            [0.0, 0.0, 1.0, 0.0],

```

```

        [0.0, 0.0, 0.0, 1.0]
    ],
    "name": "Cube",
    "data_name": "Cube.015",
    "type": "MESH"
  },
  "origin_world": [0.0, 0.0, 0.0],
  "materials": []
},

"...": "JSON content continues exactly as in the original file",

"meta": {
  "scene": "Scene",
  "units": {
    "system": "METRIC",
    "length_unit": "MILLIMETERS",
    "scale_length": 1.0
  },
  "schema_version": "3.3.6-extended-lite-roundtrip-locked-names",
  "blender_version": "4.2.1 LTS",
  "mesh_counts": {
    "vertex_count": 8,
    "face_count": 12,
    "triangle_count": 12
  }
}
}
]
}

```

References

1. Latupeirissa, J.E.; Arrang, H. Sustainability factors of building information modeling (BIM) for a successful construction project management life cycle in Indonesia. *J. Build. Pathol. Rehabil.* **2024**, *9*, 26. <https://doi.org/10.1007/s41024-023-00376-1>.
2. Gomes, A.M.; Azevedo, G.; Sampaio, A.Z.; Sánchez-Lite, A. BIM in structural projects: Interoperability analyses and data management. *Appl. Sci.* **2022**, *12*, 8814. <https://doi.org/10.3390/app12178814>.
3. Lai, H.; Deng, X. Interoperability analysis of IFC-based data exchange between heterogeneous BIM software. *J. Civ. Eng. Manag.* **2018**, *24*, 537–555. <https://doi.org/10.3846/jcem.2018.6132>.
4. Gerbino, S.; Cieri, L.; Rainieri, C.; Fabbrocino, G. On BIM interoperability via the IFC standard: An assessment from the structural engineering and design viewpoint. *Appl. Sci.* **2021**, *11*, 11430. <https://doi.org/10.3390/app112311430>.
5. Pazlar, T.; Turk, Ž. Interoperability in practice: Geometric data exchange using the IFC standard. *J. Inf. Technol. Constr.* **2008**, *13*, 362–380.
6. Sampaio, A.Z.; Gomes, A.M.; Farinha, T. BIM methodology applied in structural design: Analysis of interoperability in ARCHICAD/ETABS process. *J. Softw. Eng. Appl.* **2021**, *14*, 189–206. <https://doi.org/10.4236/jsea.2021.146012>.
7. Aranda, J.Á.; Martín-Dorta, N.; Naya, F.; Conesa-Pastor, J.; Contero, M. Sustainability and interoperability: An economic study on BIM implementation by a small civil engineering firm. *Sustainability* **2020**, *12*, 9581. <https://doi.org/10.3390/su12229581>.
8. Jiang, S.; Jiang, L.; Han, Y.; Wu, Z.; Wang, N. OpenBIM: An enabling solution for information interoperability. *Appl. Sci.* **2019**, *9*, 5358. <https://doi.org/10.3390/app9245358>.
9. Liu, Y.; Lin, H.; Zhao, Z.; Bai, W.; Hu, N. Research on the visualization of railway signal operation maintenance based on BIM + GIS. *Sensors* **2023**, *23*, 5984. <https://doi.org/10.3390/s23135984>.

10. Kaiser, C.; Jaud, Š.; Schlenger, J. B-spline surfaces in IFC: Implementing an open-source viewer. In *Proceedings of the 33rd Forum Bauinformatik*, Munich, Germany, 6–7 September 2022; pp. 33–34.
11. Edwards, G.; Li, H.; Wang, B. BIM Based Collaborative and Interactive Design Process Using Computer Game Engine for General End-Users. *Vis. Comput. Eng.* **2015**, *3*, 4. <https://doi.org/10.1186/s40327-015-0018-2>.
12. Rehman, S.U.; Kim, I.; Hwang, K.-E. Advancing BIM and game engine integration in the AEC industry: Innovations, challenges, and future directions. *J. Comput. Des. Eng.* **2025**, *12*, 26–54. <https://doi.org/10.1093/jcde/qwaf030>.
13. Semeraro, Semeraro, C.; Biyrouti, S.J.A.; Abdelkareem, M.A.; Olabi, A.G. Digital Twin-Driven Innovation in Smart and Green Building: A Structured Review and Research Agenda. *Int. J. Thermofluids* **2025**, *29*, 101393. <https://doi.org/10.1016/j.ijft.2025.101393>.
14. Wu, J.; Chen, J.; Chen, G.; Wu, Z.; Zhong, Y.; Chen, B.; Ke, W.; Huang, J. Development of Data Integration and Sharing for Geotechnical Engineering Information Modeling Based on IFC. *Adv. Civ. Eng.* **2021**, 8884864. <https://doi.org/10.1155/2021/8884864>.
15. Lee, S. Automatic Building Information Model Query and Intermediate Representation for Effective Model Exchange. *J. Inf. Technol. Constr.* **2015**, *20*, 616–634.
16. Solihin, W.; Eastman, C.; Lee, Y.-C. Multiple Representation Approach to Achieve High-Performance Spatial Queries of 3D BIM Data Using a Relational Database. *Autom. Constr.* **2017**, *81*, 369–388. <https://doi.org/10.1016/j.autcon.2017.03.014>.
17. Wang, Z.; Sacks, R.; Ouyang, J.; Ying, S.; Borrmann, A. A Framework for Generic Semantic Enrichment of BIM Models. *J. Comput. Civ. Eng.* **2024**, *38*, 4023038. <https://doi.org/10.1061/JCCEE5.CPENG-5487>.
18. Mohamed, A. Versioning of Geometry Representations in BIM Models. Master's Thesis, Technical University of Munich, Munich, Germany, 2022. Available online: <https://mediatum.ub.tum.de/1690154> (accessed on 24 December 2025).
19. Zhu, J.; Wu, P.; Chen, M.; Kim, M.J.; Wang, X.; Fang, T. Automatically Processing IFC Clipping Representation for BIM and GIS Integration at the Process Level. *Appl. Sci.* **2020**, *10*, 2009. <https://doi.org/10.3390/app10062009>.
20. Aung, N.; Choi, W.; Kulinan, A.S.; Cha, G.; Park, S. Three-Dimensional Engine-Based Geometric Model Optimization Algorithm for BIM Visualization with Augmented Reality. *Sensors* **2022**, *22*, 7622. <https://doi.org/10.3390/s22197622>.
21. Palihakkara, A.D.; Osorio-Sandoval, C.A.; Tizani, W. Review of the Application of Ontologies and Semantic Web for Building Information Modelling and Digital Twins Based Construction Management. *J. Inf. Technol. Constr.* **2025**, *30*, 1208–1243. <https://doi.org/10.36680/j.itcon.2025.049>.
22. Nugraha, B.S.; Larasati, A.; Dñng, H.C. Assessing the Interoperability and Semantic Readiness of BIM and IFC Data for AI Integration in the Architecture, Engineering, and Construction Industry: A Systematic Review. *Int. J. Intell. Data Mach. Learn.* **2025**, *2*, 14–24. Available online: <https://aimjournals.com/index.php/ijidml/article/view/344> (accessed on 24 December 2025).
23. Arumugam, M. BIM and AI Integration for Smart and Sustainable Infrastructure: A Comprehensive Review of Trends, Technologies, and Future Directions. *Int. J. Civ. Eng. Technol.* **2024**, *15*, 26–47. https://doi.org/10.34218/IJCIET_15_05_003.
24. Noardo, F.; Krijnen, T.; Ohori, K.A.; Biljecki, F.; Ellul, C.; Harrie, L.; Eriksson, H.; Polia, L.; Salheb, N.; Tauscher, H.; et al. Reference Study of IFC Software Support: The GeoBIM Benchmark 2019—Part I. *Trans. GIS* **2021**, *25*, 805–841. <https://doi.org/10.1111/tgis.12709>.
25. Lee, G.; Won, J.; Ham, S.; Shin, Y. Metrics for quantifying the similarities and differences between IFC files. *J. Comput. Civ. Eng.* **2011**, *25*, 172–181. [https://doi.org/10.1061/\(ASCE\)CP.1943-5487.0000077](https://doi.org/10.1061/(ASCE)CP.1943-5487.0000077).
26. Badenko, V.; Bolshakov, N.; Celani, A.; Puglisi, V. Principles for Sustainable Integration of BIM and Digital Twin Technologies in Industrial Infrastructure. *Sustainability* **2024**, *16*, 9885. <https://doi.org/10.3390/su16229885>.
27. Du, S.; Hou, L.; Zhang, G.; Tan, Y.; Mao, P. BIM and IFC Data Readiness for AI Integration in the Construction Industry: A Review Approach. *Buildings* **2024**, *14*, 3305. <https://doi.org/10.3390/buildings14103305>.
28. Yu, Y.; Kim, S.; Jeon, H.; Koo, B. A Systematic Review of the Trends and Advances in IFC Schema Extensions for BIM Interoperability. *Appl. Sci.* **2023**, *13*, 12560. <https://doi.org/10.3390/app132312560>.
29. Zhu, J.; Wang, X.; Wang, P.; Wu, Z.; Kim, M.J. Integration of BIM and GIS: Geometry from IFC to shapefile using open-source technology. *Autom. Constr.* **2019**, *102*, 105–119. <https://doi.org/10.1016/j.autcon.2019.02.014>.
30. Afsari, K.; Eastman, C.; Castro-Lacouture, D. JavaScript Object Notation (JSON) Data Serialization for IFC Schema in Web-Based BIM Data Exchange. *Autom. Constr.* **2017**, *77*, 24–51. <https://doi.org/10.1016/j.autcon.2017.01.011>.
31. Şenol, H.İ.; Gökgöz, T. Integration of Building Information Modeling (BIM) and Geographic Information System (GIS): A New Approach for IFC to CityJSON Conversion. *Earth Sci. Inform.* **2024**, *17*, 3437–3454. <https://doi.org/10.1007/s12145-024-01343-1>.
32. Nam, D.; Lee, D.; Lee, S.; Kwon, S. Performance comparison of 3D file formats on a mobile web browser. *Int. J. Internet Broadcast. Commun.* **2019**, *11*, 31–42. <https://doi.org/10.7236/IJIBC.2019.11.2.31>.

33. Du, J.; Shi, Y.; Mei, C.; Quarles, J.; Yan, W. Communication by Interaction: A Multiplayer VR Environment for Building Walkthroughs. In Proceedings of the Construction Research Congress 2016, San Juan, Puerto Rico, 31 May–2 June 2016; pp. 2281–2290. <https://doi.org/10.1061/9780784479827.227>.
34. Pixar Animation Studios. Universal Scene Description (USD) Overview. Available online: <https://openusd.org> (accessed on 24 December 2025).
35. Apple Inc. USD Workflows and Integration. Available online: <https://developer.apple.com/augmented-reality/> (accessed on 24 December 2025).
36. Biljecki, F.; Ledoux, H.; Stoter, J.; Vosselman, G. The Variants of an LOD of a 3D Building Model and Their Influence on Spatial Analyses. *ISPRS J. Photogramm. Remote Sens.* **2016**, *116*, 42–54. <https://doi.org/10.1016/j.isprsjprs.2016.03.003>.
37. Banfi, F. The Evolution of Interactivity, Immersion and Interoperability in HBIM: Digital Model Uses, VR and AR for Built Cultural Heritage. *ISPRS Int. J. Geo-Inf.* **2021**, *10*, 685. <https://doi.org/10.3390/ijgi10100685>.
38. Hakim, A.; van der Vaart, J.; Arroyo Otori, K.; Stoter, J. Development of a Geo to BIM Converter: CityJSON Importer Plugin for Autodesk Revit. In *Recent Advances in 3D Geoinformation Science: Proceedings of the 18th 3D GeoInfo Conference*; Kolbe, T.H., Donaubauer, A., Beil, C., Eds.; Lecture Notes in Geoinformation and Cartography; Springer: Cham, Switzerland, 2024; pp. 249–263. https://doi.org/10.1007/978-3-031-43699-4_15.
39. May, K.W.; Walsh, J.; Smith, R.T.; Gu, N.; Thomas, B.H. UnityRev—Bridging the Gap between BIM Authoring Platforms and Game Engines by Creating a Real-Time Bi-Directional Exchange of BIM Data. In *Proceedings of the 27th International Conference of the Association for Computer-Aided Architectural Design Research in Asia (CAADRIA 2022)*, Sydney, Australia, 9–15 April 2022; Volume 2, pp. 527–536.
40. Johansson, M.; Roupé, M. Real-World Applications of BIM and Immersive VR in Construction. *Autom. Constr.* **2024**, *158*, 105233. <https://doi.org/10.1016/j.autcon.2023.105233>.
41. Zaman, A.A.U.; Abdelaty, A.; Sobuz, M.H.R. Integration of BIM Data and Real-Time Game Engine Applications: Case Studies in Construction Safety Management. *J. Inf. Technol. Constr.* **2024**, *29*, 117–140. <https://doi.org/10.36680/j.itcon.2024.007>.
42. Wang, Z.; Zhang, H.; Qiang, G.; Zang, J.; Tang, S. Enhancing BIM for Construction Safety Education on Mobile Devices with Unreal Engine. *Intell. Build. Int.* **2025**. <https://doi.org/10.1080/17508975.2025.2551012>.
43. Afzal, M.; Shafiq, M.T. Evaluating 4D-BIM and VR for Effective Safety Communication and Training: A Case Study of Multi-Lingual Construction Job-Site Crew. *Buildings* **2021**, *11*, 319. <https://doi.org/10.3390/buildings11080319>.
44. Babalola, A.; Manu, P.; Cheung, C.; Yunusa-Kaltungo, A.; Bartolo, P.J.D.S. A Systematic Review of the Application of Immersive Technologies for Safety and Health Management in the Construction Sector. *J. Saf. Res.* **2023**, *85*, 66–85. <https://doi.org/10.1016/j.jsr.2023.01.007>.
45. Massafra, A.; Jabi, W.; Gulli, R. Topological BIM for Building Performance Management. *Autom. Constr.* **2024**, *166*, 105628. <https://doi.org/10.1016/j.autcon.2024.105628>.
46. Yuan, Y.; Yang, J.; Shao, R.; Mao, J.; Zhang, C. Research on BIM Model Lightweighting Methods and IoT Technology Application in the Context of WebGL. *J. Phys. Conf. Ser.* **2022**, *2215*, 12008. <https://doi.org/10.1088/1742-6596/2215/1/012008>.
47. Cho, J. Semantic-Vertex-Based Topological Detection for Automatic Dimension Generation in Building Information Modeling (BIM) with Industry Foundation Classes (IFC). *Appl. Sci.* **2026**, *16*, 139. <https://doi.org/10.3390/app16010139>.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.