Neural Implicit Representations for IFC Objects: A Coordinate-Preserving Approach for Volumetric Spatial Reasoning

Sang Du[1,2], Lei Hou[1,2] , Guomin(Kevin) Zhang[1,2] and Haosen Chen[1,2]

[1]School of Engineering, RMIT University, Melbourne, 3000, Victoria, Australia
[2]Centre for Future Construction, RMIT University, Melbourne, 3000, Victoria, Australia
sang.du@student.rmit.edu.au, lei.hou@rmit.edu.au, kevin.zhang@rmit.edu.au, haosen.chen@rmit.edu.au.

Abstract –

Recent advances in three-dimensional (3D) generative Artificial Intelligence (AI) hold significant potential for automating architectural design. However, existing methods cannot effectively process Industry Foundation Classes (IFC) data due to its complex structure. There is a pressing need for a unified representation that preserves the object-level spatial information for generative tasks. This paper proposes an auto-decoder approach that converts IFC object geometry into a unified, generation-ready representation, enabling volumetric spatial reasoning. First, an IFC export strategy that retains each object in its original global coordinates frame is introduced. Second, a sampling method that pairs 3D points with their distance to the nearest surface is applied. Third, a neural network that jointly optimises per-object representation vectors and the decoder is presented, yielding unified representations for IFC objects. Finally, an octree-based decoding method that enables efficient geometry reconstruction is employed. Comparative experiments demonstrate that the proposed method significantly outperforms voxel-based baselines in reconstruction fidelity. Furthermore, these unified latent vectors successfully enable downstream AI models to utilise IFC spatial information for complex volumetric reasoning , achieving an overall accuracy of 96.8% in classifying multi-constraint spatial interactions. This is validated by evaluating the classification of multi-constraint spatial interactions between slab openings and door clearances, which achieved an overall accuracy of 96.8%. This method converts complex geometry data into a unified representation, thereby enabling the use of IFC in generative AI.

Keywords –

BIM, IFC, 3D geometry representation, signed distance function, generative AI

1 Introduction

Three-dimensional (3D) generative Artificial Intelligence (AI) has rapidly advanced in recent years, demonstrating a remarkable ability to learn from existing data and synthesise novel 3D models [1], [2]. Particularly, in digital entertainment, it has been applied to learn and create coherent virtual worlds that reflect real-world plausibility [3], [4]. This ability to learn spatial constraints and patterns is particularly valuable for the built environment. For example, architectural designs require content to be spatially coherent [5]. 3D generative AI, therefore, holds significant potential for automating aspects of architectural design. For instance, current generative studies in architectural design still focus on generating floor plans or façade images [6], [7].

However, establishing such an AI system requires access to high-quality architectural design data [8]. The Industry Foundation Classes (IFC) standard is a vendor-neutral, ISO standardised Building Information Modelling (BIM) schema that couples precise building geometry with rich semantic attributes [9]. After two decades of global adoption, extensive design data has accumulated in IFC format [10], [11]. The vast amount of data contained in IFC models has supported a range of AI-based studies, such as intelligent construction management, automated risk assessment, automated compliance checking, and autonomous construction robotics [12], [13]. Despite the promise, current 3D generative AI applications in BIM are rarely based on IFC, thereby overlooking the enriched 3D geometry data it offers. For example, Text2BIM [14] generated 3D models using design software operation commands from a large language model, while FloorDiffusion [15] created floor plans through training on a dedicated 2D residential floor plan dataset. Although IFC natively contains the same geometric and spatial information, both applications bypass the IFC model and instead utilise other data sources.

The problem stems from the fact that 3D generative AI has difficulty processing IFC's complex geometry data directly [8]. IFC employs multiple paradigms for geometry representation, with each paradigm relying on distinct mathematical formulations and modelling procedures [9]. By contrast, 3D generative AI must satisfy data format requirements from two aspects. First, general AI learning requires a unified, fixed-dimensional data format [16]. Second, generative AI requires a data format that supports interpolation. Interpolation is the process of determining unknown values between two known values [17]. This mismatch constitutes a representation gap that prevents direct use of IFC data in 3D generative AI.

To bridge this gap, existing research in the Architecture, Engineering, Construction (AEC) domain has explored various approaches. Among the possible alternatives, 3D geometry embedding emerges as a promising solution. This method converts 3D geometry into unified, fixed-length vectors for AI use while preserving fine-grained geometric detail. Broadly, such methods can be grouped into explicit and implicit embeddings [18]. Explicit embeddings are effective for discriminative tasks such as classification, but do not emphasise generative aspects, whereas implicit embeddings achieve more substantial generative performance. In this study, we focus on signed distance function (SDF) based implicit embeddings, which describe a shape with points in space and its distance to the shape [19]. This method yields a unified, fixed-length 3D geometry representation that is suitable for generative tasks. However, existing SDF-based methods typically normalise the object's spatial information to facilitate shape learning [20]. In the built environment, maintaining spatial coherence requires careful preservation of fundamental object properties such as shape, size, location, and orientation [5]. Consequently, the loss of spatial information limits the applicability of existing SDF methods to generative tasks, such as automated architectural design.

To bridge the representation gap between IFC data and 3D generative AI, this study presents an auto-decoder approach that converts IFC object geometry into a unified, generation-ready representation while preserved spatial information. Our contributions include:

- Proposing an SDF-based auto-decoder that embeds IFC geometry into a unified representation for AI utilisation, preserving accurate spatial information and thereby enabling volumetric spatial reasoning.
- Developing a complete BIM-to-representation pipeline that automates data preparation, model training, and result visualisation, thereby enabling practical deployment.

2 Literature Review

3D geometry embedding is a method that turns 3D shapes into fixed-length vectors. These vectors allow AI models to process geometric data easily. Researchers group these methods into explicit and implicit categories. Each category has different strengths and weaknesses for building design tasks.

Explicit embedding methods encode the visible parts of an object. One common method is multi-view projection [21], [22]. It turns 3D elements into sets of 2D images. This approach uses simple data formats. However, it discards important volumetric and topological information. Another method uses point clouds [23], [24], [25]. Point cloud encoders capture a shape's global structure. But these encoders are sensitive to noise and uneven sampling. They also lack explicit connections between points. Voxel-grid methods divide 3D space into regular cubes [26], [27]. These grids work well with standard 3D neural networks. But they have low resolution because of high memory costs. Fine geometric details are often lost in voxel grids. Mesh encoders work directly with vertices and faces [28], [29], [30]. They keep high-quality surfaces and topology. However, their performance depends on the quality of the mesh triangulation. Generally, explicit methods are well-suited for tasks such as classification. Their reconstruction quality is usually limited by the resolution of the input data.

Implicit methods represent 3D objects as continuous fields in space. They use neural networks to approximate these fields. This produces compact codes that enable high-quality reconstruction. Signed Distance Function (SDF) methods are a popular choice [20], [31], [32]. They learn the distance from any point in space to the object's surface. This helps the model capture very small variations in shape. Neural Radiance Fields (NeRFs) generate realistic images from different viewpoints [33], [34]. But their geometry is hidden within density values. They do not align well with the specific parts of a building model. Occupancy networks decide if a point is inside or outside an object [35], [36]. They can handle a wide range of shapes and topologies.

A significant problem exists with current implicit methods. Most of these methods use a step called normalisation. They move every object to the centre of a small unit cube. This step helps the AI learn general shapes more easily. But it removes the building's real-world scale and position. Architectural design needs this spatial context. Designers must know the exact location of parts to check joints and assembly rules. Normalised methods cannot distinguish between different interfaces or adjacent components.

To address these critical limitations, it is essential to develop a representation that maintains spatial integrity while providing high-quality visual information to support human judgment. This study utilises an SDF-based auto-decoder to overcome the deficiencies of existing implicit methods by preserving the original spatial setting and project coordinates of components. In addition to ensuring spatial accuracy, this approach enables high-fidelity 3D reconstruction for direct visual inspection. By delivering both precise coordinates and viewable geometry, the proposed method makes building data significantly more robust and intuitive for AI-driven design tasks.

3 Methodology

The methodology described in the sources presents a comprehensive pipeline for transforming Industry Foundation Classes (IFC) 3D geometric data into AI-compatible vector representations. This approach, implemented as a web-based tool, focuses on capturing the rich spatial and semantic context of architectural components to support downstream AI-driven design tasks. The pipeline consists of four integrated stages as in Figure 1: component extraction, spatial sampling, auto-decoder implementation, and geometric reconstruction. Unlike conventional computer vision methods that often strip away real-world metadata, this framework is specifically designed to operate within the original project coordinate frame, preserving scale, position, and orientation throughout the process.

The decision to bypass traditional unit cube normalisation is driven by the inherent position-dependent nature of architectural design. While standard normalisation facilitates shape learning in general computer vision, it discards the absolute elevation and relative topological relationships, such as the specific clearance between a door and a slab opening, that are fundamental to BIM. In large-scale models, maintaining geometric consistency across heterogeneous components is vital. If global coordinates are lost, downstream tasks like automated clash detection and volumetric spatial reasoning become mathematically impossible. Therefore, preserving the original project coordinate frame is a necessary design choice to ensure that the AI-driven reasoning reflects real-world construction constraints and spatial logic.

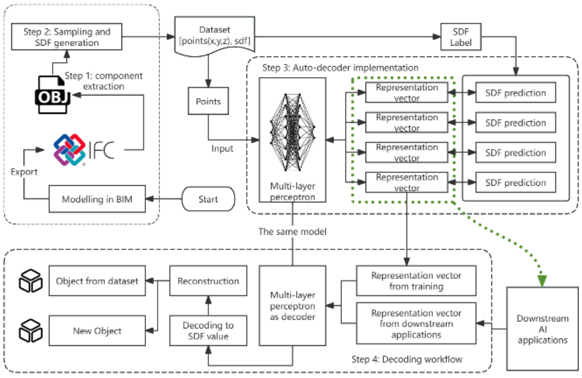

Figure 1. Overall architecture of the proposed method

The first stage, component extraction, involves isolating primary architectural entities—such as walls, columns, beams, slabs, doors, and windows—from the IFC model. Because neural representations based on SDFs require watertight, closed surfaces, the extracted geometry is converted to OBJ meshes and repaired where necessary. To handle the complexity of heterogeneous modelling practices, the workflow employs geometric proxy substitution. For instance, complex multi-part assemblies, such as windows and doors, are replaced with simplified cuboid proxies that preserve their functional volumes and spatial bounds. A critical aspect of this stage is the application of a 0.1 global scale factor, which serves as a linear unit change to ensure that coordinates and distance values remain within a stable numerical range for deep learning optimisation. Most importantly, the original coordinates and orientations are stored alongside these meshes, distinguishing this method from standard pipelines that normalise objects into an abstract unit cube and lose their real-world context.

Following extraction, the sampling and SDF generation phase converts the watertight meshes into training data. This process utilises a two-stage, surface-guided strategy to ensure the neural network learns both the precise shape of the component and the "free space" surrounding it. In the first stage, points are sampled uniformly across the object's surface to ensure total coverage. In the second stage, local jittering is applied around these surface points to create a dense cloud of near-field and far-field samples. Each sampled 3D point is paired with its SDF value, which represents the shortest distance to the component's surface; negative values indicate the interior, and positive values indicate the exterior. By sampling up to a 10-metre radius in real-world coordinates, the model is explicitly supervised to recognise that no geometry exists in the surrounding envelope, effectively preventing the AI from "hallucinating" surfaces in empty regions.

The core of the technical framework is the auto-decoder implementation. Unlike traditional encoder-decoder architectures that require an explicit network to generate features, the auto-decoder directly optimises object-specific representation vectors alongside the neural network weights during training. The architecture is based on a Multilayer Perceptron (MLP) with eight layers and 512 hidden units per layer. The network takes a 128-dimensional shape vector and a 3D spatial coordinate as input and predicts the corresponding SDF value. By sharing the same network weights across a family of similar architectural components, the model is forced to encode geometric variations, such as changes in window size or column thickness, within the latent vector space. This structured latent space enables the model to generalise across unseen architectural designs while maintaining the integrity of the spatial relationships learned from the training data.

The final stage is the reconstruction process, which identifies the zero-level set (the boundary where the SDF value is zero) to recover the explicit 3D surface from the learned vectors. Since searching a large spatial domain for these boundaries is computationally expensive, the methodology employs an adaptive octree refinement strategy. This hierarchical approach begins with a coarse grid and recursively subdivides space into smaller blocks only in regions near the predicted surface, reaching a maximum depth of five levels. Areas identified as "empty" or far from the surface are pruned to save memory and processing time. Finally, the Marching Cubes algorithm is applied to generate a triangulated mesh from the identified zero-crossing points. This bidirectional conversion ensures that the compact neural representations can be translated back into standard 3D formats for visual inspection and integration into conventional architectural modelling workflows. The following experiment validates the method using a common spatial coordination problem.

4 Experiment

4.1 Comparative Analysis with Voxel-based Representation

To justify the selection of the SDF-based auto-decoder, we conducted a comparative experiment against 3D-VAE, a representative voxel-based variational auto-encoder. A subset of 50 wall and slab components from the IFCNet dataset was utilised for evaluation. To ensure a fair comparison of representation capacity, both methods were tested under normalised conditions.

As summarised in Table 1, the proposed implicit representation achieves reconstruction errors approximately one order of magnitude lower than the voxel baseline across Chamfer distance, Hausdorff distance, and surface normal error. The qualitative comparison of these results is illustrated in Figure 2, which depicts representative reconstructions for wall and slab components.

Table 1. Reconstruction errors of the proposed method vs. 3D-VAE voxel baseline.

Scenario	Ours vs GT	3D-VAE vs GT
Chamfer Distance (mm, mean)	2.26	12.45
Hausdorff Distance (mm, mean)	8.17	54.87
Surface Normal Error (deg, mean)	7.65	19.54

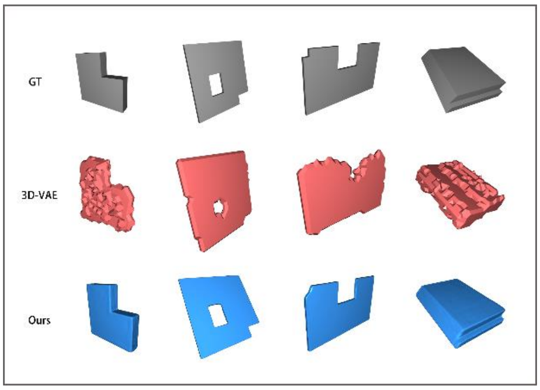

Figure 2. Qualitative comparison of per-object reconstructions. From top to bottom: ground-truth (GT), the 3D-VAE voxel baseline, and ours.

The visual evidence in Figure 2 demonstrates that the implicit representation better preserves sharp profiles and clean surfaces. By contrast, the voxel-based baseline exhibits significant aliasing and boundary artifacts due to its fixed resolution. These results indicate that while voxelization offers faster end-to-end processing, its sensitivity to resolution limits its use for high-fidelity IFC design checking. The proposed implicit representation yields smoother, more faithful surfaces, which is essential for preserving spatial integrity in BIM-to-AI workflows.

4.2 Downstream AI demonstration

To verify that the latent vectors produced by the proposed IFC-to-SDF pipeline can be used directly by downstream AI models, we designed a small demonstration experiment. The objective is to show that a simple neural network can learn implicit design rules solely from latent vectors and perform cross-component, multi-constraint design checking. The test case is derived from a common architectural design scenario involving the coordination of building services and structural elements. Vertical service penetrations in the floor slab require specific openings, particularly for water pipes, which must be carefully positioned relative to nearby wall features. Potential conflicts arise when door swings overlap with these service openings and access routes. Furthermore, openings placed too close to the base of a wall can violate the minimum spatial clearances required for the structural integrity and precise installation of adjacent components. This scenario represents a typical cross-component, multi-constraint interaction, where traditional rule-based design checks often struggle to account for all possible spatial variations. By contrast, SDF-based latent vectors enable data-driven learning of these complex relationships, as they represent 3D objects as continuous fields that capture both the geometry and the surrounding "free space". This approach allows the neural network to identify spatial constraints without relying on handcrafted rules or explicit geometric coordinates.

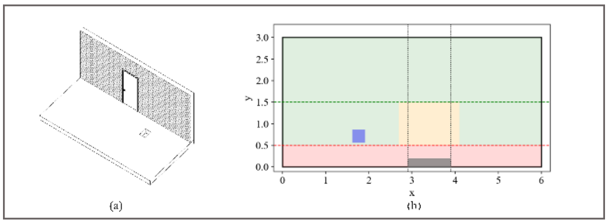

Figure 3. (a) Revit model of the slab–wall–door configuration used in the experiment. A rectangular vertical opening is placed on the slab surface. (b) Corresponding plan-view representation used for SDF-based rule classification. The red region marks the clash zone (objects with $y < 0.5$ m), the yellow region denotes the near-clearance band around the door in the mid-height range, and the green area represents the safe-far zone. The blue rectangle indicates the position of the opening sampled for this instance.

The illustrative geometry, shown in Figure 3(a), consists of a single slab, a single wall, and one door. We define three design rules for any given door position on the wall, as illustrated in Figure 3(b). In the global y direction, we measure the distance from the nearest edge of the floor opening to the wall. When the distance is less than 0.5 m, the configuration is labelled as a spatial clash case and shown in red. When the distance exceeds 1.5 m, the configuration is labelled as safe. For intermediate cases where the y-direction distance lies between 0.5 m and 1.5 m, an additional horizontal condition is applied. We check whether the horizontal projections of the door and opening overlap. We also measure their separation along the x direction. When the projections do not overlap, and their separation exceeds 0.2 m, the configuration is labelled as safe and shown in green. All remaining cases are labelled as attention required cases and shown in yellow. The yellow region indicates cases where the horizontal projections overlap, or the separation is less than or equal to 0.2 m. Such cases require additional design judgment.

The dataset for the experiment is generated procedurally. The door height is fixed at 2.1 m. The door width is sampled uniformly between 0.8 m and 1.2 m, yielding 50 distinct door variants with random positions along the wall. The floor opening size ranges from 0.2 m × 0.2 m to 0.6 m × 0.6 m. We sampled 50 distinct opening variants at random locations on the slab. All 50 door variants are combined with all 50 opening variants, yielding 2,500 unique door and opening configurations. Each configuration is labelled automatically according to the three design rules described above, producing three classes of outcomes: clash, near-clearance, and safe. The labelled dataset is then split into training, validation, and test subsets with a 0.7/0.15/0.15 ratio.

To test whether the latent vectors capture the relevant spatial relationships, we use a simple MLP classifier. The input consists only of the slab's latent codes, including its opening and the door. We adopt a two-tower architecture in which one MLP branch processes the door latent vector, and the other processes the slab latent vector. The outputs of the two branches are concatenated and passed through additional fully connected layers to predict the three design classes. After training, the classifier achieves high accuracy on the held-out test set. Representative quantitative results are reported in Table 2. Qualitative examples of correctly classified configurations are shown in Figure 4.

Table 2. Classification performance of the slab–door rule-learning classifier on the test set.

Label	Total sample	Correct predictions	Accuracy (%)
Safe	273	268	98.2
Near clearance	59	52	88.1
Clash	45	45	100.0
Overall	377	365	96.8

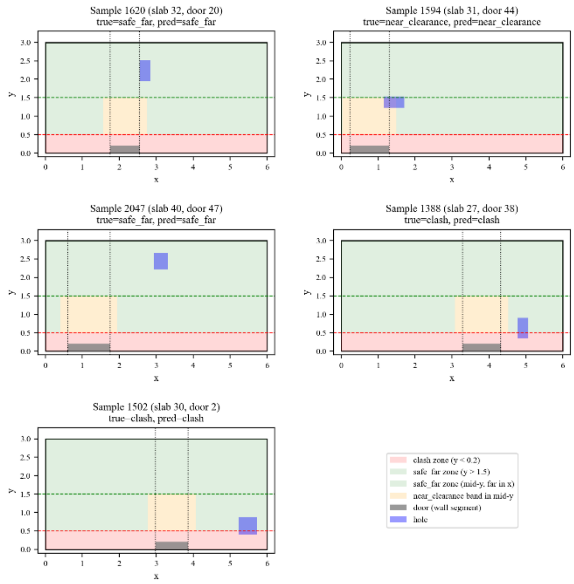

Figure 4. Representative plan-view samples from the slab–door dataset showing the rule-based labelling and MLP predictions. Each subplot depicts a slab with a vertical hole (blue) and a door segment along the wall (grey). The red band marks the clash zone ($y < 0.5$), the light green region marks the safe_far zone ($y > 1.5$), and the mid-height band is split into a near_clearance region (yellow) and a safe_far region when the door is sufficiently far in x. Vertical dashed lines indicate the x-gap thresholds used in the labelling rules. Titles report the sample index and slab/door IDs, together with the ground-truth label and the model prediction, illustrating that the classifier recovers the intended geometric rules for these examples.

The experimental results indicate that a downstream MLP can learn the spatial relationships encoded in the latent vectors. The model discovers the geometric correlation between door and slab opening positions without access to explicit geometric coordinates or handcrafted features. The model generalises well to unseen combinations of door and opening geometries and achieves high classification accuracy for new spatial configurations. In practical applications, similar pipelines can be used when as-built building projects are available as IFC models. The pipeline can encode building components and their penetrations into latent vectors and train AI models that learn from past adjustment decisions made to satisfy actual building construction. Such trained models could then provide data-driven feedback on new building designs, supporting designers with limited construction experience in checking clearances and avoiding cross-component conflicts.

5 Discussion: Pathways to Generative AI

The neural representation proposed in this study establishes a foundational framework that provides a technical bridge to Generative AI within the BIM domain. This application methodology follows a systematic pipeline that transforms irregular architectural data into a machine-learnable format. By mapping discrete IFC entities into a continuous feature space, the framework enables advanced generative architectures to interact effectively with complex structural information.

The first stage of this process involves standardising heterogeneous data through feature encoding to overcome the inherent irregularities of IFC geometries. Our approach utilises an auto-decoder architecture to compress diverse building entities into a series of fixed-length latent vectors. This transformation is essential because it converts complex symbolic data into a uniform format that mainstream machine learning models can process efficiently.

The coordinate-preserving nature of the resulting latent space serves as a critical enabler for various generative tasks. Since spatial positioning and geometric features are embedded directly within the vector distribution, generative models such as Diffusion Models or Generative Adversarial Networks can be trained to recognise underlying spatial assembly rules. These models operate directly within the latent space to predict optimal layouts or suggest design iterations while maintaining strict adherence to the global coordinate system.

Finally, the trained neural decoder functions as a restorative bridge back to the physical domain once a new latent vector is generated or predicted. The decoder interprets the synthesised vector and reconstructs it into a high-fidelity 3D geometric model. Because the latent space is conditioned on precise coordinate data, the reconstructed entities automatically maintain their intended spatial orientations and topological relationships. This closed-loop methodology ensures that AI-generated outputs remain fully compatible with standard BIM workflows and engineering requirements.

6 Conclusion

This study addresses the representation gap between complex IFC geometry and the data needs of AI-driven automated architectural design. The researchers developed an SDF-based auto-decoder that converts building components into compact latent vectors while preserving their original spatial context. Experiments show that the method is accurate enough for automated design coordination. Preserving real-world coordinates is essential, as standard normalisation results in significant accuracy loss. Additionally, the model supports the generation of design variants through smooth interpolation of size, shape, and position. While the current experiment focuses on rectangular shapes and faces computational challenges for large projects, it successfully bridges the gap between IFC entities and neural representations. By isolating variables in a controlled setting, the study demonstrates the fundamental validity of coordinate preservation. Future research will focus on improving efficiency, conducting more robust evaluations on complex real-world scenarios, and creating new datasets to help AI learn complex assembly rules despite geometric noise.

References

[1] L. Qiu *et al.*, "Richdreamer: A generalizable normal-depth diffusion model for detail richness in text-to-3d," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 9914–9925. doi: 10.1109/CVPR52733.2024.00946.

[2] A. Sengupta, T. Alldieck, N. Kolotouros, E. Corona, A. Zanfir, and C. Sminchisescu, "Diffhuman: Probabilistic photorealistic 3d reconstruction of humans," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 1439–1449. doi: 10.1109/CVPR52733.2024.00143.

[3] K. Sargent *et al.*, "Zeronvs: Zero-shot 360-degree view synthesis from a single image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 9420–9429. doi: 10.1109/CVPR52733.2024.00900.

[4] Y. Yang *et al.*, "Holodeck: Language guided generation of 3d embodied ai environments," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024, pp. 16227–16237. doi: 10.1109/CVPR52733.2024.01536.

[5] P. Zierke, "The History of spatial coherence," *Zeszyty Naukowe Politechniki Poznańskiej. Architektura, Urbanistyka, Architektura Wnętrz*, 2023, doi: 10.21008/j.2658-2619.2023.17.25.

[6] G. Luo *et al.*, "Controllable and flexible residential floor plan layout design based on multi-agent deep reinforcement learning with layout prior size and similar experience abandon," *Advanced Engineering Informatics*, vol. 68, p. 103702, 2025, doi: 10.1016/j.aei.2025.103702.

[7] M. Deng, V. J. Gan, Y. Tan, A. Joneja, and J. C. Cheng, "Automatic generation of fabrication drawings for façade mullions and transoms through BIM models," *Advanced Engineering Informatics*, vol. 42, p. 100964, 2019, doi: 10.1016/j.aei.2019.100964.

[8] S. Du, L. Hou, G. Zhang, Y. Tan, and P. Mao, "BIM and IFC Data Readiness for AI Integration in the Construction Industry: A Review Approach," *Buildings*, vol. 14, no. 10, p. 3305, 2024, doi: 10.3390/buildings14103305.

[9] BuildingSMART International, "Industry Foundation Classes (IFC4) – ISO 16739-1:2018," International Organization for Standardization, 2018. [Online]. Available: https://standards.buildingsmart.org/IFC/RELEASE/IFC4/ADD2_TC1/HTML/

[10] Y. Pan and L. Zhang, "Integrating BIM and AI for smart construction management: Current status and future directions," *Archives of Computational Methods in Engineering*, vol. 30, no. 2, pp. 1081–1110, 2023, doi: 10.1007/s11831-022-09830-8.

[11] R. Safour, S. Ahmed, and B. Zaarour, "BIM Adoption around the World," *International Journal of BIM and Engineering Science*, vol. 4, no. 2, pp. 49–63, 2021, doi: 10.54216/IJBES.040202.

[12] G. Chen, A. Alsharef, A. Ovid, A. Albert, and E. Jaselskis, "Meet2Mitigate: An LLM-powered framework for real-time issue identification and mitigation from construction meeting discourse," *Advanced Engineering Informatics*, vol. 64, p. 103068, 2025, doi: 10.1016/j.aei.2024.103068.

[13] D. Chen, L. Chen, Y. Zhang, S. Lin, M. Ye, and S. Sølvsten, "Automated fire risk assessment and mitigation in building blueprints using computer vision and deep generative models," *Advanced Engineering Informatics*, vol. 62, p. 102614, 2024, doi: 10.1016/j.aei.2024.102614.

[14] C. Du, S. Esser, S. Nousias, and A. Borrmann, "Text2BIM: Generating building models using a large language model-based multi-agent framework," *arXiv preprint arXiv:2408.08054*, 2024, doi: 10.48550/arXiv:2408.08054.

[15] J. Shim, J. Moon, H. Kim, and E. Hwang, "FloorDiffusion: Diffusion model-based conditional floorplan image generation method using parameter-efficient fine-tuning and image inpainting," *Journal of Building Engineering*, vol. 95, p. 110320, 2024, doi: 10.1016/j.jobe.2024.110320.

[16] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013, doi: 10.1109/TPAMI.2013.50.

[17] W. H. Press, *Numerical recipes 3rd edition: The art of scientific computing*. Cambridge University Press, Cambridge, 2007.

[18] O. Poursaeed, M. Fisher, N. Aigerman, and V. G. Kim, "Coupling explicit and implicit surface representations for generative 3d modeling," in *European Conference on Computer Vision*, Springer, 2020, pp. 667–683. doi: 10.1007/978-3-030-58607-2_39.

[19] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, 1996, pp. 303–312. doi: 10.1145/237170.237269.

[20] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "DeepSDF: Learning Continuous Signed Distance Functions for Shape Representation," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA: IEEE, Jun. 2019, pp. 165–174. doi: 10.1109/CVPR.2019.00025.

[21] L. Xu, X. Feng, J. Liu, H. Qi, and Y. Wang, "Automatic defect localization method based on BIM projected image," *Advanced Engineering Informatics*, vol. 69, p. 103874, 2025, doi: 10.1016/j.aei.2025.103874.

[22] X. Zhou *et al.*, "BIM product recommendation for intelligent design using style learning," *Journal of Building Engineering*, vol. 73, p. 106701, 2023, doi: 10.1016/j.jobe.2023.106701.

[23] A. Nichol, H. Jun, P. Dhariwal, P. Mishkin, and M. Chen, "Point-e: A system for generating 3d point clouds from complex prompts," *arXiv preprint arXiv:2212.08751*, 2022, doi: 10.48550/arXiv:2212.08751.

[24] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660. doi: 10.1109/CVPR.2017.16.

[25] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph cnn for learning on point clouds," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 5, pp. 1–12, 2019, doi: 10.1145/3326362.

[26] A. Brock, T. Lim, J. M. Ritchie, and N. Weston, "Generative and discriminative voxel modeling with convolutional neural networks," *arXiv preprint arXiv:1608.04236*, 2016, doi: 10.48550/arXiv.1608.04236.

[27] D. Maturana and S. Scherer, "Voxnet: A 3d convolutional neural network for real-time object recognition," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Ieee, 2015, pp. 922–928. doi: 10.1109/IROS.2015.7353481.

[28] Y. Siddiqui *et al.*, "Meshgpt: Generating triangle meshes with decoder-only transformers," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2024, pp. 19615–19625. doi: 10.1109/CVPR52733.2024.01855.

[29] R. Hanocka, A. Hertz, N. Fish, R. Giryes, S. Fleishman, and D. Cohen-Or, "Meshcnn: a network with an edge," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–12, 2019, doi: 10.1145/3306346.3322959.

[30] Y. Feng, Y. Feng, H. You, X. Zhao, and Y. Gao, "Meshnet: Mesh neural network for 3d shape representation," in *Proceedings of the AAAI conference on artificial intelligence*, 2019, pp. 8279–8286. doi: 10.1609/aaai.v33i01.33018279.

[31] C. Xu, M. Mielle, A. Laborde, A. Waseem, F. Forest, and O. Fink, "Exploiting semantic scene reconstruction for estimating building envelope characteristics," *Building and Environment*, vol. 275, p. 112731, 2025, doi: https://doi.org/10.1016/j.buildenv.2025.112731.

[32] P. Wang, L. Liu, Y. Liu, C. Theobalt, T. Komura, and W. Wang, "Neus: Learning neural implicit surfaces by volume rendering for multi-view reconstruction," *arXiv preprint arXiv:2106.10689*, 2021, doi: 10.48550/arXiv:2106.10689.

[33] M. Tancik *et al.*, "Block-nerf: Scalable large scene neural view synthesis," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2022, pp. 8248–8258. doi: 10.1109/CVPR52688.2022.00807.

[34] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng, "Nerf: Representing scenes as neural radiance fields for view synthesis," *Communications of the ACM*, vol. 65, no. 1, pp. 99–106, 2021, doi: 10.48550/arXiv.2003.08934.

[35] J. Chibane, T. Alldieck, and G. Pons-Moll, "Implicit functions in feature space for 3d shape reconstruction and completion," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 6970–6981. doi: 10.1109/CVPR42600.2020.00700.

[36] L. Mescheder, M. Oechsle, M. Niemeyer, S. Nowozin, and A. Geiger, "Occupancy networks: Learning 3d reconstruction in function space," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 4460–4470. doi: 10.1109/CVPR.2019.00459.