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

FROM GENERAL SPACE PLANNER TO LEARNING-BASED GENERATION: A SYSTEMATIC MAPPING OF REPRESENTATION, OBJECTIVES, AND CONTROL

Suhyung Jang[1,2] and André Borrmann[1,2]
[1]Georg Nemetschek Institute, Technical University of Munich, Munich, Germany
[1]Chair of Computing in Civil and Building Engineering, School of Design and Engineering, Technical University of Munich, Munich, Germany

Abstract

Recent advances in learning-based artificial intelligence (AI) show that shared representations (e.g., pixelized images and tokenized corpora) enable multitask foundation model training, whereas digital building layout representations have diverged. Grounded in Eastman's General Space Planner, this study systematically maps 59 studies over a half-century (1971–2025) to trace how building layout representations evolved with expanding objectives and emerging control mechanisms. Publications surged in the final half-decade (52.5%), with representations shifting from simplified to free-formed representations as control mechanisms and objectives diversify. The trends suggest a bottleneck, as consistent transfer of learning-based building layout generation is difficult without a shared representation.

Introduction

Automated building layout planning has been studied for more than five decades, beginning with early computational approaches in the 1960s and evolving through successive waves of constraint-based reasoning (Liggett, 2000), optimization, and, more recently, pattern-based generation (Weber et al., 2022). Despite such sustained effort, automatically generated building layouts have often been limited to single-floor rectilinear primitives. Although recent image-generation approaches have expanded the variety of generated forms, they still face difficulties integrating with current computer-aided design (CAD) or building information modeling (BIM) workflows (Jang et al., 2025b).

A building layout is not merely a set of space boundaries. Spaces, together with adjacency, connectivity, containment, and other relations must be considered. In addition, layout representations should reflect the hierarchy of building components (e.g., levels, rooms, and building elements). These requirements become more pronounced as layout planning shifts from conforming to connectivity and planned proportions conditions to satisfying multiple and sometimes competing design objectives such as code compliance, target cost, environmental performance, and user preferences.

Digital building layout representation, therefore, must represent geometry and topology simultaneously and often must combine multiple encodings to do so (Jang et al., 2025a). As a result, the field has progressed through diverse and sometimes incompatible choices of representation, alongside diversified objective formulations, emerging representations and control techniques. This has made it difficult to establish a stable representational basis for comparing methods or transferring models across building layout tasks. By contrast, in vision and language, shared representations provide stable minimal units for composition spaces, allowing learning-based artificial intelligence (AI) models to learn common structures and generalize across datasets and downstream tasks (LeCun et al., 2015).

For raster images, inputs are consistently defined by a two-dimensional pixel grid with channel values, and for text, contemporary transformer pipelines convert language to token sequences whose semantics are embedded into latent spaces to model context. These shared conventions have been central to multitask learning and foundation models. Building layout representations, however, have diverged over time as objectives and control mechanisms expanded, leaving unclear which representational components are minimally required for transferable learning-based generation across diverse objectives and contexts.

To address this issue, this paper analyzes the automated building layout planning studies conducted over the last half-century to clarify how building layouts have been represented, what has been optimized, and how layout generation has been controlled. To make longitudinal comparisons possible, we use Charles Eastman's General Space Planner (GSP) as a foundational reference (Eastman, 1973), and reorganize its core elements into a three-part analytical framework consisting of representation, objective, and control for decade-by-decade trend mapping (Figure 1).

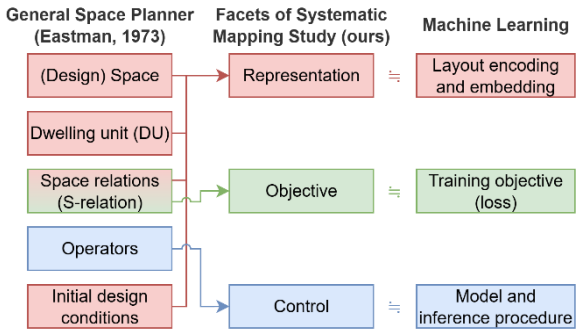

Figure 1: Three analytical facets of the systematic mapping study

Representation captures how layouts are encoded in terms of spatial units, boundaries, and relational structures. Objectives capture what requirements are targeted and how they are formalized through metrics, constraints, or performance criteria. Control captures how planning proceeds through mechanisms used to generate, modify,

and converge on candidate layouts, including search, optimization, constraint solving, learning-based procedures, or interactive refinement. In generalized learning-based building layout generation, these three facets correspond to the representational space in which layouts are encoded, the objective function that defines desirable solutions, and the procedural mechanism through which candidate layouts are generated and refined. This framing is designed to provide a common analytical basis for comparing classical and recent learning-based approaches within a single coding scheme. The goal of this study is to systematically identify persistent elements as central features in building layout planning over multiple decades, as well as to elucidate emerging requirements and methodological advancements associated with the increasing complexity of building layout planning. The analysis is directed toward establishing prerequisites for learning-based AI models that support transferable and extensible general space planning.

The remainder of this paper is organized as follows. The second section describes the systematic mapping study protocol and coding procedure. The third, fourth, and fifth sections present decade-by-decade analyses of trends in representation, objectives, and control. The sixth section discusses current challenges and future prospects, and the seventh section concludes the paper.

Research method

Systematic mapping studies aim to structure a research area by classifying and counting contributions across a predefined scheme, providing an overview of what has been studied and where the literature has appeared. We adopted the updated systematic mapping study guidelines (Petersen et al., 2015), and followed a protocol consisting of planning, conducting, and reporting activities.

Study identification

Space layout planning has co-evolved with broader layout optimization research in manufacturing and industrial engineering, which makes keyword-only retrieval prone to substantial noise. To keep the map focused on building layout planning, a discipline-scoped sampling strategy was adopted for study identification. This study first identified studies from representative journals in the field by discipline. Journal names were collected from Scimago Journal Rankings (SJR), gathering all journals under the "Engineering" subject area, specifically "Building and Construction," and "Architecture" subject categories. Then we developed a Python code using pybliometrics.scopus, a library that supports application programming interface (API)-based searching, to collect the papers from the Scopus database within the relevant disciplines. We queried Scopus using title-level layout terms, namely "space planning," "space layout," "building layout," "architectural layout," "room layout," "space plan," "room plan," "floor plan," and "floorplan," combined with building-related filters in titles, abstracts, and keywords (e.g., "building," "architectural," "dwelling unit," "room," and "housing"). In total, 212 records were retrieved within the target time window (1970-2025).

Study selection was conducted in two stages through title and abstract screening, followed by a full-text screening. We excluded (1) review papers, (2) design theory papers without layout generation, (3) layout analysis papers without automated planning/generation, and (4) studies focusing on urban or site-scale layouts. This screening resulted in 35 included articles.

To improve coverage of earlier and transitional work, we additionally identified candidate studies from two prior secondary sources covering the search window, with one source targeting studies before 2000 (Liggett, 2000) and another targeting studies between 2000 and 2025 (Xu et al., 2025). After removing duplicates and records that could not be retrieved in full text, 17 additional articles were included. Additionally, seven conference papers from the computer vision and machine learning domain were included after snowballing. In total, 59 records were included in the final map.

We then organized the 59 included studies by decade for analysis. The final corpus contains 3 studies from the 1970s, 2 from the 1980s, 6 from the 1990s, 5 from the 2000s, 12 from the 2010s, and 31 from 2020–2025, meaning that publications from the most recent five years account for 52.54% of the screened corpus.

Data extraction and classification

For systematic mapping, we constructed a facet-based classification scheme grounded in GSP's core concepts and aligned the facets with learning-based AI components to support cross-paradigm comparison.

GSP defines a layout planning problem using the following concepts: Space, Dwelling Unit (DU), Space Relations (S-relations), Operators, and an initial design condition. In GSP, S-relations specify constraints to be satisfied, and operators specify how DUs are manipulated within Space. However, across fifty years of research, the role of "relations" has expanded: relations are used in two ways: (1) to encode relational structure (e.g., adjacency / connectivity graphs) and (2) to define what must be satisfied or optimized (e.g., requirements, objectives, performance criteria). To avoid conflating these roles, we separate relation information into two coding targets: relational encoding under *representation*, and relation-derived requirements under *objectives*.

The classification is organized into three primary facets: *representation*, *objective*, and *control*:

- *Representation* captures how the design space, units, and relations are encoded. Because building layouts are often represented using mixed forms (e.g., rectangular vector boundaries together with adjacency graphs), we record (R1) the primary representation and (R2) one or more supporting representations. We also map the representation dimension (R3), where 2.5D augments planar layouts with limited height information (e.g., extrusion or vertical elements for simulation) and 3D models spaces as full volumetric entities.

- *Objective* captures what requirements the method targets and how they are formalized. We record (O1) objective structure (e.g., constraint satisfaction, single-objective, multi-objective, non-optimization generation, learned objective) and (O2) one or more objective metric families (e.g., connectivity, cost, preferences, performance).
- *Control* captures how candidate layouts are generated, modified, and converged. We record (C1) the primary control mechanism, (C2) supporting mechanisms used for enforcement or refinement, and (C3) the interaction mode.

Table 1: Sub-facets and categories of the three analytical facets of systematic mapping study.

Facet	Sub-facet	Categories
Representation (R)	Primary representation (R1)	R1∈{Vector (Rectangle), Vector (Coordinate), Graph, Grid, Voxel, Image}
	Supporting representation (R2)	R2⊆{Vector (Rectangle), Vector (Coordinate), Graph, Grid, Voxel, Image}
	Dimension (R3)	R3∈{2D, 2.5D, 3D}
Objectives (O)	Structure (O1)	O1∈{Constraint-satisfaction, Single-objective, Multi-objective, Learned-objective, Non-objective-oriented}
	Metric (O2)	O2⊆{Adjacency/Connectivity, Area/Size, Circulation/Travel, Distance/Proximity, Orientation/View, Thermal/Energy, Lighting/Daylight, Code/Regulation, Cost, Structural/Engineering performance, Preference/User, Constructability}
Control (C)	Primary mechanism (C1)	C1∈{Operation-based, Case-based, Parameter-based, Learned-based (single-stage), Learned-based (multi-stage)}
	Supporting mechanism (C2)	C2⊆{Operation-based, Case-based, Parameter-based, Learned-based (single-stage), Learned-based (multi-stage)}
	Interactive control (C3)	C3∈{Automatic generation, Interactive generation}

The classification scheme was developed iteratively through an initial category draft based on domain knowledge and subsequent refinement by consolidating and renaming categories based on patterns observed in the collected studies (Table 1).

Representation (R)

Representation describes how a building layout is encoded as a computational object, including the spatial primitives (e.g., rooms, boundaries, cells) and the relational structures used to express adjacency, connectivity, and hierarchy when applicable. Because layout planning requires both geometric reasoning (shape, size, coordinates) and topological reasoning (relations and constraints), many studies rely on heterogeneous representational stacks rather than a single universal format. Consistent with early space-planning work that combined relational encodings with explicit geometric units (Eastman, 1973; Grason, 1971; Pfefferkorn, 1972), we map representation at three levels: (1) a primary representation (R1) that serves as the main modeling unit for generation or optimization, (2) a supporting representation (R2) used to encode constraints, semantics, or downstream structures, and (3) the representation dimension (R3), which captures whether the layout is modeled in 2D, 2.5D, or 3D. Here, planar layout generation with extrusion but without explicit vertical relations is classified as 2.5D, while methods that represent vertical relations are classified as 3D.

Primary representation (R1)

Figure 2 summarizes a shift from early emphasis on coarse digital layout encodings toward higher geometric freedom via grid- and image-based formats.

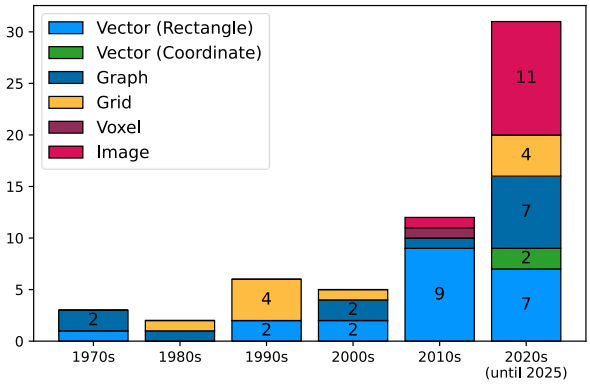

Figure 2: Period-wise distribution of primary representation formats (R1).

In the 1970–80s (n=5), graph dominates (60.0%), with vector-rectangle and grid each at 20.0%, reflecting topology-oriented formulations with simple geometric instantiation (Roth et al., 1982). Grid-based representation marks an early move toward freer boundary expression (Liggett and Mitchell, 1981). In the 1990s (n=6), grid-based representations become the majority primary representation (66.7%). Several studies in this period proposed operator-based grid methods that expand space boundaries to satisfy constraints and produce an initial draft, which is then refined by a human designer toward a less constrained layout (Jo and Gero, 1995). In the 2000s (n=5), primary encodings split between graph and vector-rectangle (40.0% each), with grid at 20.0%, indicating coexistence of topological representation and explicit geometric units. The 2010s (n=12) are dominated by vector-rectangle (75.0%) while image and voxel begin to appear (8.3% each), reflecting diversification through both 2D and 3D encodings (Dino, 2016; Hua, 2016; Marson and Musse, 2010). In the 2020s (until 2025; n=31), image becomes the largest primary category (35.5%), with graph still substantial (22.6%). Overall, the trend indicates increasing reliance on representations that support flexible boundaries and forms, while graph persists as a stable carrier of topological representation.

Supporting representation (R2)

Figure 3 reports supporting representations used alongside R1. Since use of R2 is optional, the total number reflects coded occurrences rather than the number of papers per period. In the 1970–80s (n=4), vector-rectangle dominates (75.0%) with graph at 25.0%, consistent with setups where topological reasoning is primary but explicit rectangular geometry is used to instantiate spatial units. In later work, an explicit separation between topology and geometry is made more systematic; for example, Medjdoub and Yannou (2000) generate a graph describing topological configuration first and then place rectangular space units to realize geometry. In the 2010s (11 coded occurrences), graph becomes the dominant supporting representation (81.8%), with vector-coordinate and vector-rectangle each at 9.1%, suggesting more frequent use of graph as a feasibility/semantic scaffold around diverse primary encodings. In the 2020s (until 2025; n=29), graph remains the most frequent supporting representation (51.7%), followed by image (24.1%) and vector-rectangle (20.7%), with grid at 3.4%. This aligns with recent pipelines that combine image generation with explicit relational structure; for instance, Zheng and Petzold (2023) show graph-guided generation of topology-reflected layout images by using the graph together with image representations. Across periods, even as boundary representation becomes freer (grid/image/vector-coordinate), graph-based support remains central for preserving or recovering adjacency relations and enforcing topological constraints.

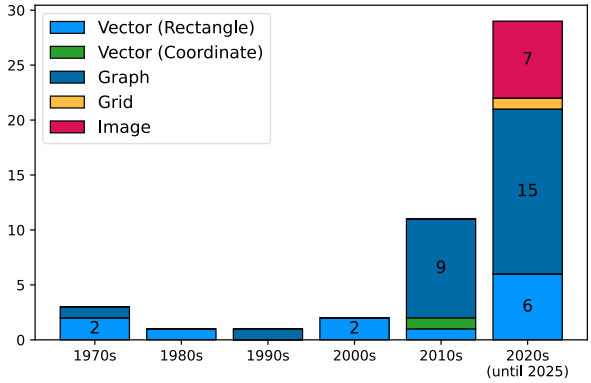

Figure 3: Period-wise distribution of supporting representation formats (R2).

Representation dimension (R3)

Figure 4 indicates that most studies remain focused on 2D representation, with limited adoption of 2.5D or 3D layout generation. In the 1970–80s (n=5), 1990s (n=6), and 2000s (n=5), all coded cases are 2D (100%). In the 2010s (n=12), 2D remains the majority (75.0%), while 2.5D appears (16.7%) and explicit 3D appears in a small minority (8.3%). The only 3D case uses voxel-based representation (Dino, 2016). In the 2020s (until 2025; n=31), 2D remains dominant (90.3%) with 2.5D at 9.7%. Most 2.5D cases arise when planar layouts are extruded or augmented with vertical elements (e.g., walls/windows) to support downstream simulation goals. For example, Rodrigues et al. (2014) extruded generated floor plans to form an energy-performance simulation loop for automated layout generation. Overall, fully automated 3D building layout generation remains underexplored, even though many studies note vertical variation in real buildings as a key limitation and future direction.

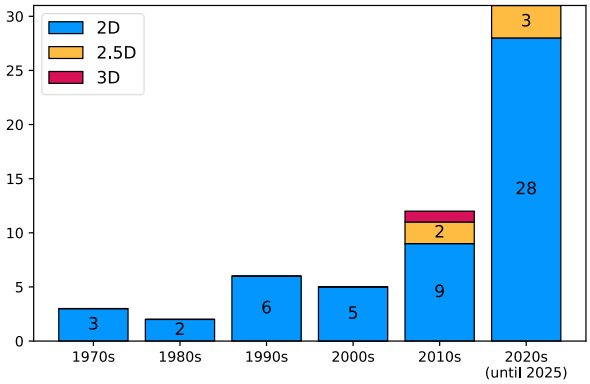

Figure 4: Period-wise distribution of representation dimension (R3).

Objectives (O)

Objectives capture what layout planning methods attempt to achieve and how requirements are formalized into computational targets. To remain compatible across rule-based optimization and learning-based pipelines, we separate objectives into objective structure (O1), which describes the formal type of target, and objective metrics (O2), which describe the substantive design criteria being operationalized.

Objective structure (O1)

Figure 5 shows a transition from constraint-satisfaction-based generation to multi-criteria optimization and, more recently, learned loss formulations. In the 1970–80s (n=5), objective structure is dominated by constraint satisfaction (80.0%), with the remainder framed as single-objective optimization (20.0%), reflecting early feasibility and rule-compliance formulations. From the 1990s (n=6), multi-objective optimization becomes dominant (66.7%), while constraint satisfaction remains substantial (33.3%), consistent with the broader use of evolutionary and annealing-style optimization for competing criteria.

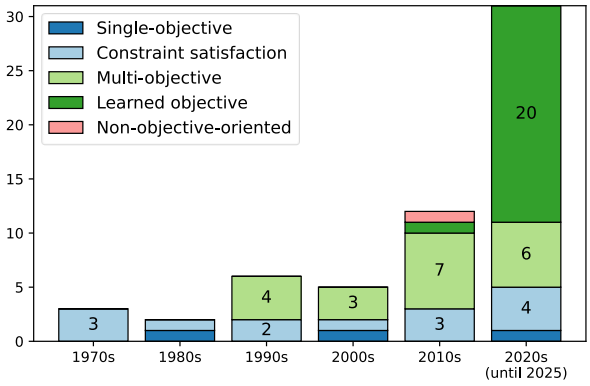

Figure 5: Period-wise distribution of objective structure (O1).

In the 2000s (n=5), multi-objective remains the majority (60.0%), with constraint satisfaction and single objective

each at 20.0%. In the 2010s (n=12), multi-objective continues to lead (58.3%), while constraint satisfaction accounts for 25.0%, and both learned objective (i.e., loss minimization) and non-optimization generation appear (8.3% each), indicating early entry of learning-based formulations. In the 2020s (until 2025; n=31), learned objective (loss minimization) becomes dominant (64.5%), while multi-objective persists (19.4%) and constraint satisfaction remains present (12.9%), with single objective at 3.2%. Overall, as learning-based AI becomes increasingly central to automated building layout generation, objective attainment shifts toward learning optimized patterns from accumulated data, which in turn elevates the importance of representations that can robustly encode and map design-relevant structure (e.g., geometry and topology) into machine-comprehensive formats.

Objective metric (O2)

Figure 6 indicates continuity in core building layout factors and divergence in the objective metrics over time. Across all periods, adjacency/connectivity and area/size are the most persistent and frequently utilized objectives, indicating that they are minimal requirements for functional building layout planning.

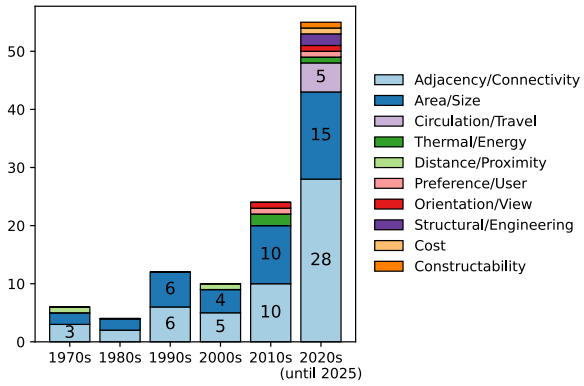

Figure 6: Period-wise distribution of objective metric categories (O2).

In early periods, between 1970s to the 2000s, are mainly concentrated on adjacency/connectivity, area/size, and distance/proximity conformance. Through the 2010s, additional categories appear, including thermal/energy, preference/user, and orientation/view. Between 2020-2025, the coded metric families expand further, adding circulation/travel, structural/engineering, cost, and constructability. These expanding objective scopes are represented by several examples. Rodrigues et al. (2014) exemplifies thermal- and energy-driven optimization by generating plan layouts under topological constraints and searching for alternatives that improve simulated thermal/energy performance. Likewise, Almashaqbeh and El-Rayes (2021) is representative of work optimizing cost and constructability, particularly in modular layout contexts. This diversification of more implementation-specific objectives reflects increasing interest beyond adjacency and area constraints satisfaction.

Control (C)

Control captures how building layout generation proceeds (i.e., the mechanisms used to explore, modify, and converge on candidate solutions) in each study. This facet includes classical strategies such as operator-based search and optimization (e.g., rule-based geometric operations followed by evaluation), case-based reuse (e.g., cased-based reasoning), parameter-based exploration of a defined design space (e.g., parametric modeling and design parameter exploration), and recent learning-based generation (e.g., generative adversarial networks, diffusion models, and transformer models). We code control into primary control mechanism (C1), supporting mechanism (C2), and interaction type (C3).

Primary control mechanism (C1)

Figure 7 shows a transition from operation-centric generation to learned generation in the 2020s, while non-learned mechanisms remain present. In the 1970–80s (n=5), primary control is entirely operation-based (100.0%). In the 1990s (n=6), operation-based control remains dominant (83.3%), with case-based approaches appearing at 16.7%. In the 2000s (n=5), operation-based and parameter-based control each account for 40.0%, with case-based at 20.0%, indicating diversification of classical control strategies. In the 2010s (n=13), operation-based control returns to a strong majority (69.2%), with parameter-based at 15.4%, and learned multi-stage and case-based each at 7.7%. In the 2020s (until 2025; n=31), learned single-stage becomes the largest category (41.9%), followed by learned multi-stage (32.2%), while operation-based (n=5) and parameter-based (n=3) remain at 16.1% and 9.7%. This split suggests that multi-stage learned pipelines may represent a modern reconfiguration of earlier operation-centric control, where learning provides the initial synthesis and operation-style steps reappear as gradual refinement to enforce feasibility and relational consistency.

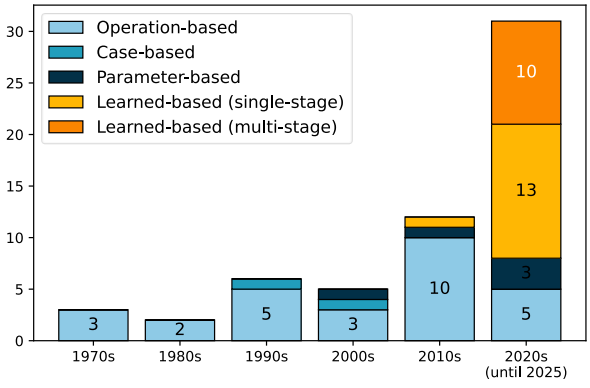

Figure 7: Period-wise distribution of primary control mechanisms (C1).

Supporting control mechanism (C2)

Figure 8 depicts that the use of C2 is sparsely reported across the mapped studies, so interpretation is restricted to cases where a supporting mechanism is explicitly described. In the 1990s, the observed supporting mechanisms are entirely parameter-based (3 occurrences).

In the 2000s, operation-based supporting control appears only once (1 occurrence). In the 2010s, parameter-based supporting control is more common (3 occurrences) while operation-based refinement again appears only in an isolated instance (1 occurrence). In the 2020s (until 2025 in this coding), the observed supporting mechanisms are parameter-based (2 occurrences), with no operation-based supporting refinement recorded.

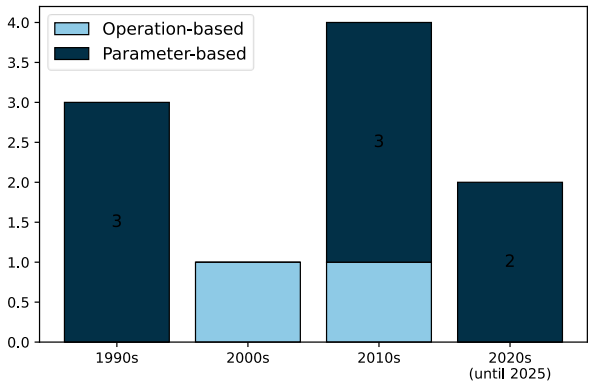

Figure 8: Period-wise distribution of supporting control mechanisms (C2).

Taken together, C2 functions as a robustness layer that stabilizes outcomes on top of the primary control mechanism. Parameter-based supporting control is consistent with restricting or re-parameterizing the design space to reduce sensitivity and guide convergence, whereas operation-based supporting control aligns with hybrid workflows where predefined operators are applied as cleanup or feasibility enforcement after an initial non-operator step. This pattern is consistent with pipelines where a layout is first proposed by a higher-level procedure and then adjusted through operators until constraints and geometric realizability are satisfied. For example, Rosenman (Rosenman, 2000) describes an evolutionary approach in which candidate plans are iteratively transformed and then modified with operators until constraints are met. Merrell et al. (2010) similarly illustrate a learned-to-operator pattern, where a probabilistic model supports program/adjacency generation and axis-aligned split/trim operators instantiate and refine rectangular layout, using supporting steps to ensure feasibility and geometric realization.

Interactive control (C3)

Across the field, interaction is typically limited to setting initial requirements (e.g., constraints, program, adjacency) while generation proceeds automatically, and interactive refinement remains comparatively underrepresented as a core research setting (Figure 9). Recent work begins to incorporate explicit interactive control modes: Shekhawat et al. (2021) propose a system where a user edits an adjacency graph via a graphical user interface (GUI) and the system generates a grid-based plan satisfying the edited relations; Zeng et al. (2024) similarly assume a designer-provided connectivity/adjacency graph and generate a plan layout conditioned on that structure. These examples indicate a shift toward interactive injection of topology or constraints in the 2020s, but the dominant workflow remains fully automated generation.

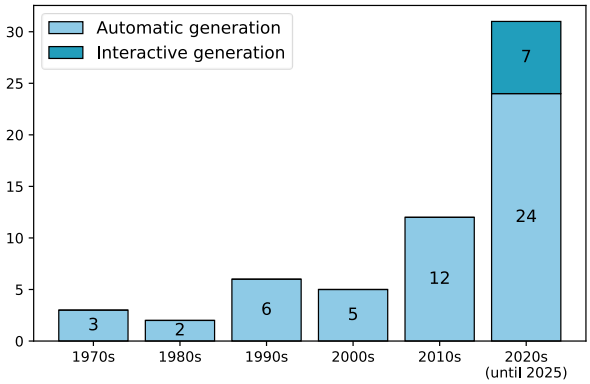

Figure 9: Period-wise distribution of interactive control types (C3).

Discussion

Over the last half-century, representation choices in building layout planning have diversified alongside increasing objective complexity and more sophisticated control mechanisms. At the same time, several primitives remain consistently central, most notably spatial proportions and adjacency relationships. The most recent half-decade marks a quantitative and qualitative inflection driven by learning-based AI. The rise of image-based primary representations and loss-minimization objectives, and learning-based control have become more prominent, increasing geometric freedom while further elevating the importance of representation and inductive bias in model learning.

Yet the trends observed in this study suggest that the field is moving toward divergence rather than convergence, unlike the trajectory that enabled foundation models in the natural language and vision domains. The confusion matrix in Figure 10 highlights this diverging landscape of representations, with the primary representation on the y-axis and the secondary representation on the x-axis. Early work in the 1970s primarily combined graph-based topology with rectangular vector units. In the 1980s–1990s, grid-based representations became more prominent, offering a robust design space that can express diverse shapes while preserving relative spatial relations. From the 2000s into the 2010s, hybridization became more systematic to represent richer context: rectangular primitives often remain the main modeling unit, while graphs increasingly serve as supporting scaffolds to maintain topological consistency.

In the 2020s, representation combinations diversify sharply around image-oriented generation, with graphs and vectors frequently retained as supporting encodings, and occasional pipelines that invert the direction by using images to support graph-based procedures. Overall, literature increasingly pursues higher geometric freedom while continuing to preserve explicit topological structure. This divergence may become more consequential as the field expands beyond predominantly two-dimensional layouts. As the R3 (Dimension) analysis indicates, most studies still operate in 2D, with only limited attention to

2.5D and 3D. However, real buildings are often multi-floor, and vertical spatial relationships must be accounted for, including inter-floor circulation, stacked adjacencies, and the continuity of cores and shafts. This may partly explain why 3D studies remain limited: compared with 2D layouts, 3D settings make it harder to define shared spatial units, build reusable datasets, and evaluate results under common criteria. One possible direction is to combine graph-based topology with more flexible geometric decoders, so that explicit relations can be preserved without overly constraining 3D form.

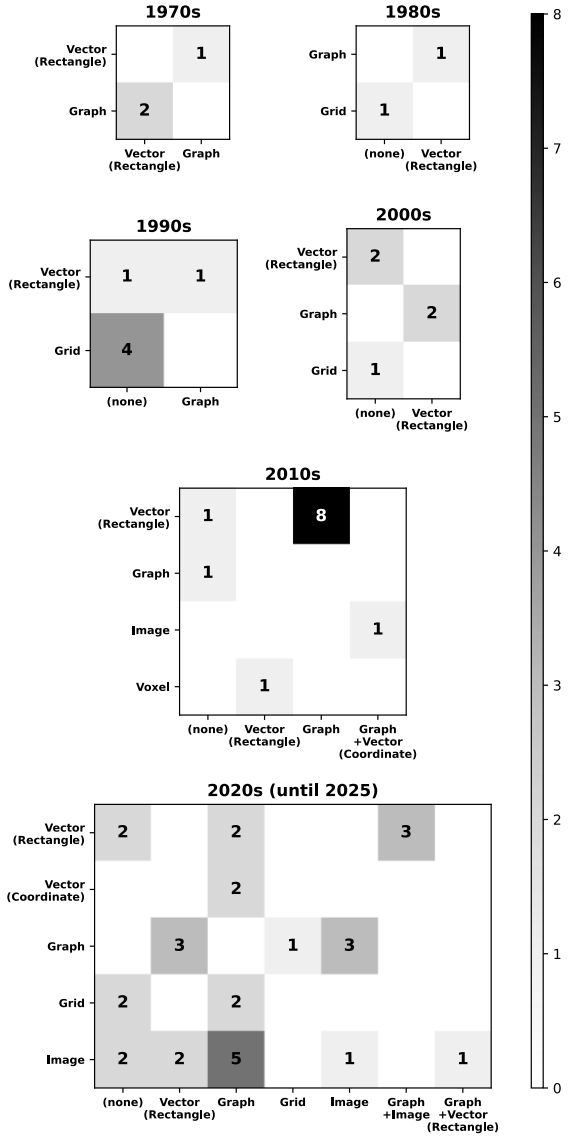

Figure 10: Combination of representation methods across periods.

If this divergence continues, the field may accumulate separate representational tracks for 2D, 2.5D, and 3D, each with different assumptions about basic units and primary relations. That would make comparison, reuse, and cumulative development more difficult. It would also weaken the conditions for domain-specific foundation model training, which usually benefits from a shared representation with stable primitives. A more convergent direction could support dataset sharing, common evaluation, and pretraining across layout tasks and dimensional settings. For practice, this means that multi-floor layout systems need representations that support not only generation, but also inspection and revision. For future benchmarks, evaluation should move beyond single-floor generation and include vertical connectivity, cross-floor consistency, and topology-preserving edits. With the expected shift toward three-dimensional building layout generation, the next stage of the field will depend in part on how this balance between geometric freedom and explicit structure is resolved.

Conclusions

This paper presented a systematic mapping study of automated building layout planning over more than five decades (1971 to 2025). Using Eastman's General Space Planner as a stable conceptual reference, we coded 59 studies along three facets: representation, objectives, and control, to enable decade-by-decade comparison across both classical and learning-based paradigms. The resulting map provides a longitudinal account of what has remained stable in layout planning, what has diversified, and where bottlenecks arise as learning-based methods become dominant.

Several findings are consistent across the full-time window. Adjacency or connectivity and area or size persist as the most central objectives, suggesting that relational coherence and proportional allocation remain minimal requirements across paradigms. As objective sets expand beyond feasibility toward broader criteria such as performance, cost, and constructability, representation and control choices co-evolve, with studies increasingly adopting richer representational stacks and more complex control strategies.

The map also indicates a recent inflection in the last half-decade, which accounts for more than half of the corpus. Learning-based generation becomes a dominant driver, coinciding with the rise of image-based primary representations and learned loss minimization. While these shifts support greater geometric freedom, they do not eliminate the need for explicit structure: graph-based encodings remain central, often as supporting representations for preserving or recovering topology and enabling feasibility and constraint enforcement. Overall, the field is trending toward representational divergence rather than convergence, unlike domains that benefited from shared conventions for foundation-model training. This divergence becomes more consequential under dimensional expansion, since most studies remain 2D despite the multi-floor nature of real buildings and the need to represent vertical relations. Continued fragmentation across 2D, 2.5D, and 3D will further raise the cost of reuse and comparison, whereas full convergence may improve comparability but risks impractical complexity when attempting to preserve both geometric freedom and explicit topology within a single formalism.

Rather than prescribing a single universal representation for building layouts, this study highlights a recurring

pattern in the literature. The pattern shows that geometric freedom is increasingly pursued through flexible encodings, while explicit topological structure is repeatedly retained. Looking ahead, especially toward three-dimensional and multi-floor layout generation, clearer reporting of representational stacks and their mappings, together with evaluation procedures that remain valid across representational and dimensional regimes, will be important for cumulative progress.

Acknowledgments

This research was supported by the National Research Foundation of Korea (NRF) grant funded by the Ministry of Science and ICT (Grant RS-2025-00522484).

References

Almashaqbeh, M., El-Rayes, K., 2021. Optimizing the modularization of floor plans in modular construction projects. J. Build. Eng. 39, 102316. https://doi.org/10.1016/j.jobe.2021.102316

Dino, I.G., 2016. An evolutionary approach for 3D architectural space layout design exploration. Autom. Constr. 69, 131–150. https://doi.org/10.1016/j.autcon.2016.05.020

Eastman, C.M., 1973. Automated space planning. Artif. Intell. 4, 41–64. https://doi.org/10.1016/0004-3702(73)90008-8

Grason, J., 1971. An approach to computerized space planning using graph theory, in: Proceedings of the 8th Design Automation Workshop, DAC '71. Association for Computing Machinery, New York, NY, USA, pp. 170–178. https://doi.org/10.1145/800158.805070

Hua, H., 2016. Irregular architectural layout synthesis with graphical inputs. Autom. Constr. 72, 388–396. https://doi.org/10.1016/j.autcon.2016.09.009

Jang, S., Borrmann, A., Lee, G., 2025a. Automated Natural Language Building Descriptor For Building Information Models, in: Proceedings of 2025 European Conference on Computing in Construction (EC3).

Jang, S., Roh, H., Lee, G., 2025b. Generative AI in architectural design: Application, data, and evaluation methods. Autom. Constr. 174, 106174. https://doi.org/10.1016/j.autcon.2025.106174

Jo, J.H., Gero, J.S., 1995. A Genetic Search Approach to Space Layout Planning. Archit. Sci. Rev. 38, 37–46. https://doi.org/10.1080/00038628.1995.9696774

LeCun, Y., Bengio, Y., Hinton, G., 2015. Deep learning. Nature 521, 436–444. https://doi.org/10.1038/nature14539

Liggett, R.S., 2000. Automated facilities layout: past, present and future. Autom. Constr. 9, 197–215.

Liggett, R.S., Mitchell, W.J., 1981. Optimal space planning in practice. Comput.-Aided Des. 13, 277–288. https://doi.org/10.1016/0010-4485(81)90317-1

Marson, F., Musse, S.R., 2010. Automatic Real-Time Generation of Floor Plans Based on Squarified Treemaps Algorithm. Int. J. Comput. Games Technol. 2010, 1–10. https://doi.org/10.1155/2010/624817

Medjdoub, B., Yannou, B., 2000. Separating topology and geometry in space planning. Comput.-Aided Des. 32, 39–61. https://doi.org/10.1016/S0010-4485(99)00084-6

Merrell, P., Schkufza, E., Koltun, V., 2010. Computer-generated residential building layouts. ACM Trans. Graph. 29, 1–12. https://doi.org/10.1145/1882261.1866203

Petersen, K., Vakkalanka, S., Kuzniarz, L., 2015. Guidelines for conducting systematic mapping studies in software engineering: An update. Inf. Softw. Technol. 64, 1–18. https://doi.org/10.1016/j.infsof.2015.03.007

Pfefferkorn, C.E., 1972. The Design Problem Solver, A System for Designing Equipment or Furniture Layouts.

Rodrigues, E., Gaspar, A.R., Gomes, Á., 2014. Automated approach for design generation and thermal assessment of alternative floor plans. Energy Build. 81, 170–181. https://doi.org/10.1016/j.enbuild.2014.06.016

Rosenman, M., 2000. Case-based evolutionary design. Artif. Intell. Eng. Des. Anal. Manuf. 14, 17–29. https://doi.org/10.1017/S0890060400141022

Roth, J., Hashimshony, R., Wachman, A., 1982. Turning a graph into a rectangular floor plan. Build. Environ. 17, 163–173. https://doi.org/10.1016/0360-1323(82)90037-3

Shekhawat, K., Upasani, N., Bisht, S., Jain, R.N., 2021. A tool for computer-generated dimensioned floorplans based on given adjacencies. Autom. Constr. 127, 103718. https://doi.org/10.1016/j.autcon.2021.103718

Weber, R.E., Mueller, C., Reinhart, C., 2022. Automated floorplan generation in architectural design: A review of methods and applications. Autom. Constr. 140, 104385. https://doi.org/10.1016/j.autcon.2022.104385

Xu, Z., Jha, N., Mehadi, S., Mandal, M., 2025. Automatic floor plan analysis: Datasets, methods, and applications (2000–2025). Autom. Constr. 178, 106378. https://doi.org/10.1016/j.autcon.2025.106378

Zeng, P., Gao, W., Yin, J., Xu, P., Lu, S., 2024. Residential floor plans: Multi-conditional automatic generation using diffusion models. Autom. Constr. 162, 105374. https://doi.org/10.1016/j.autcon.2024.105374

Zheng, Z., Petzold, F., 2023. Neural-guided room layout generation with bubble diagram constraints. Autom. Constr. 154, 104962. https://doi.org/10.1016/j.autcon.2023.104962