

Article

Digital Paradigms in Architecture: Toward a Layered Computational Ecology from Early Computation to Artificial Intelligence

Ana Isabel Lima Pacheco ^{1,2,3,*} , Isabel Clara Neves da Rocha Marques ^{1,2,3} and Ricardo Jorge Gonçalves Rocha ³

¹ CIAUD-UPT—Branch of CIAUD Research Center, Department of Architecture and Multimedia Gallaecia, Portucalense University, Rua Dr. António Bernardino de Almeida, 541, 4200-072 Porto, Portugal; isabelclaraneves@mail.upt.pt

² CIAUD, Research Centre for Architecture, Urbanism and Design, Lisbon School of Architecture, Universidade de Lisboa, Rua Sá Nogueira, Polo Universitário do Alto da Ajuda, 1349-063 Lisbon, Portugal

³ DAMG, Department of Architecture and Multimedia Gallaecia, Portucalense University, 4200-072 Porto, Portugal

* Correspondence: analima@upt.pt

Abstract

This article examines the evolution of computational paradigms in architecture through the articulation of a diachronic framework and a comparative analytical matrix. Moving beyond linear narratives centred on technological progress, the study proposes an interpretation of architectural computation as a layered ecology in which distinct regimes—symbolic, representational, informational, generative, and probabilistic—interact simultaneously. Based on a critical review of historical, theoretical, and technical sources, the study comparatively examines five major paradigmatic moments in the development of architectural computation. Instead of proposing these paradigms as discrete or sequential stages, the article interprets them as interdependent computational layers that continue to coexist within contemporary architectural practice. The findings indicate that the transition from rule-based deterministic systems to learning-based systems introduces a fundamental shift in the nature of architectural computation, moving design processes from controlled execution toward probabilistic exploration. In this context, artificial intelligence does not merely extend existing technical capabilities but reconfigures the relationships between designer, tool, and knowledge. The article concludes that contemporary architecture operates within a layered computational ecology in which multiple paradigms overlap and interact. This perspective allows computation to be understood not only as a set of tools but as an epistemological infrastructure that profoundly transforms architectural practice, its processes, and its critical frameworks.



Academic Editor: Avi Friedman

Received: 15 April 2026

Revised: 20 May 2026

Accepted: 30 May 2026

Published: 3 June 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article

distributed under the terms and

conditions of the [Creative Commons](https://creativecommons.org/licenses/by/4.0/)

[Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: digital paradigms; architectural computation; Building Information Modeling (BIM); parametric design; algorithmic design; data-driven design; design automation; layered computational ecology

1. Introduction

Over recent decades, architecture has been profoundly reconfigured by the acceleration of digitalisation processes and by the increasing integration of computational technologies into design, production, and the management of the built environment [1–3]. What initially emerged as a set of tools aimed at automating calculation and graphic representation has

progressively evolved into a complex computational ecosystem, in which algorithmic design approaches [4], parametric modelling systems [5,6], information-based environments such as Building Information Modelling (BIM) [7], and, more recently, artificial intelligence techniques [8,9] actively intervene in the formulation of architectural design.

This transformation cannot be reduced to the mere introduction of new technical instruments; rather, it entails a structural shift in the projectual process itself. Drawing is no longer understood as the graphic translation of a predefined intention, but increasingly operates as a relational system based on rules, parameters, and data-driven interactions [5–7,10]. Within this framework, the architect progressively assumes the role of author of processes and systems rather than of closed, singular objects, signalling an epistemological displacement that challenges fundamental notions of authorship, control, decision-making, and creativity [4,5].

Despite the extensive body of research dedicated to digital, computational, and algorithmic architecture, much of the existing literature remains fragmented, often focusing on specific technologies, isolated historical moments, or particular formal outcomes [2,3,11]. Less attention has been given to long-term historical and critical syntheses capable of articulating continuities and ruptures between early post-war computational paradigms and contemporary practices grounded in data-driven and learning-based systems [3,12]. As a result, the broader cultural and disciplinary implications of computation in architectural design remain insufficiently integrated into a coherent analytical framework. To address this gap, the study adopts a comparative and diachronic interpretive framework designed to analyse long-term continuities, transformations, and interdependencies across computational paradigms in architecture.

This article addresses this gap by proposing a critical and comparative framework for analysing the long-term evolution of computational paradigms in architecture. The study interprets digital development as a sequence of isolated technological substitutions, the study examines how distinct computational regimes—from early experimental systems to parametric design [5,6], Building Information Modelling [7], digitally driven design and production processes [13], and artificial intelligence and machine learning in design processes [9]—continue to coexist, overlap, and interact within contemporary architectural practice. In doing so, the article also engages broader debates concerning automation, computational agency, and the future trajectories of architectural computation [12].

In doing so, this article contributes: (i) a comparative analytical matrix synthesising five major computational paradigms in architecture, and (ii) the notion of a “layered computational ecology” as a framework for understanding the coexistence and interdependence of computational systems beyond linear narratives of technological replacement. In this study, the notion of “computational ecology” refers to a condition in which multiple computational paradigms coexist through relations of interoperability, infrastructural dependency, operational overlap, and cumulative integration, forming heterogeneous computational environments in which distinct computational logics remain simultaneously operative.

Rather than approaching technology as a neutral or deterministic force, this study frames computation as a cognitive and cultural infrastructure that actively shapes architectural knowledge and practice [11,12]. By situating digital tools within their historical and epistemological genealogy, the article contributes to a more nuanced understanding of architectural computation and provides a critical basis for assessing the implications of artificial intelligence and contemporary computational systems for the future of the discipline.

2. Theoretical Background: Computation and Architecture

The theoretical foundations of computation and computer architecture can be understood within the broader evolution of logical and mathematical thought. While Aristotelian logic represents an initial milestone in the formalisation of reasoning, it was George Boole's development of Boolean algebra that enabled the representation and manipulation of logical operations in binary form [14]. This formalisation became the mathematical basis for digital circuits and contemporary computational processing [15].

This development was further advanced by pioneers such as Leibniz and Pascal, who conceived mechanical devices for performing arithmetic operations, and, more decisively, by Charles Babbage and Ada Lovelace [16]. Their collaboration anticipated the principle of programmable machines capable of executing general sequences of instructions. Babbage's designs for the Difference Engine and the Analytical Engine, articulated with Lovelace's algorithmic formulation, introduced, for the first time, the concept of automating calculations through the application of abstract formal rules [17]. This conceptual framework was later revisited in the twentieth century, underpinning the emergence of computer-aided design and subsequently influencing architectural design methodologies [10].

In 1936, Alan Turing made a decisive contribution to the formalisation of computation theory through his paper *On Computable Numbers, with an Application to the Entscheidungsproblem*, which introduced the abstract model of the *a-machine* (automatic machine), later known as the Turing Machine [18]. Turing established the formal limits of computability, demonstrating that any effective algorithmic procedure can be expressed as a finite sequence of elementary operations on symbols, while simultaneously proving the unsolvability of the Entscheidungsproblem. The development of the first computers in the post-Second World War period, as well as Turing's later formulation of the "Imitation Game" (1950), shifted the discourse from mechanical calculation to the prospect of intelligent behaviour in artificial systems [19]. This theoretical shift contributed to the conceptual foundations of later developments in Artificial Intelligence [20].

During this period, the appearance of large-scale electronic computers such as the ENIAC (1945), developed by John Eckert and John Mauchly, demonstrated the computational capacity to address highly complex mathematical and scientific problems [21]. In parallel, programmable electromechanical machines such as the ASCC/Harvard Mark I (1944), designed by Howard Aiken in collaboration with IBM, confirmed the feasibility of sequentially controlled computational systems [22]. These technological advances established the conceptual and operational foundations that, decades later, supported the integration of digital technologies into architectural design and representation.

This convergence became particularly evident in initiatives such as the Architecture and the Computer conference (Boston Architectural College, 1965) [23], and subsequently in the Architecture Machine Group at MIT (1967), led by Nicholas Negroponte, which developed the first experimental systems of computer-aided design (CAD) [13].

The intersection of computation and architecture became increasingly pronounced with the development of early interactive graphic systems, notably Ivan Sutherland's Sketchpad (1963) [24]. This system introduced real-time geometric manipulation through the use of a light pen, implemented formal parameterisation mechanisms, and established a new paradigm of human-computer interaction supported by intuitive visual interfaces [25]. Subsequent projects, including URBAN II, URBAN V, and GENERATOR—developed within the MIT Architecture Machine Group [13]—as well as Frei Otto's investigations for the Munich Olympic Park, integrating physical and computational processes [26], demonstrated that the computer could function not only as a representational tool but also as a generative and analytical system.

As a result, a conceptual framework emerged in which digital design is understood as an algorithmic process based on formal rules, parametric relations, and computational operations [27]. This conceptual framework established the conditions for the emergence of multiple computational paradigms in architecture, including CAD, BIM, and parametric design [4–7]. Rather than replacing one another, these systems progressively accumulated and interacted, forming increasingly interconnected computational environments that later enabled the integration of artificial intelligence into architectural practice [8,9,12].

3. Methodology

This article adopts a qualitative, historical-critical methodology, situated within the constructivist paradigm (Groat & Wang) [28], based on the assumption that modes of computation in architecture are cultural, technical, and epistemological constructs rather than neutral technological evolutions. The approach is neither empirical nor quantitative; it focuses on the conceptual and interpretive analysis of digital technologies applied to architectural design. This interpretive positioning aligns with historiographic and theory-driven research traditions in architectural scholarship [10,11].

The research is structured as a systematizing review [29]. The review process combined structured bibliographic searches with theoretical sampling strategies appropriate to qualitative historical-critical research. Literature was identified through searches conducted in Scopus (Elsevier B.V., Amsterdam, Netherlands), Web of Science (Clarivate, London, United Kingdom), Google Scholar (Google LLC, Mountain View, CA, USA), and major architectural publishers using combinations of keywords such as “architectural computation”, “CAD”, “BIM”, “parametric design”, “algorithmic architecture”, “artificial intelligence in architecture”, and “computational design theory”. Selection criteria privileged publications that played a significant theoretical, historical, or operational role in the development of computational paradigms in architecture, including foundational theoretical texts, historically influential projects or systems, and widely cited disciplinary contributions. Sources focused exclusively on software implementation, purely technical optimisation, or unrelated engineering applications were excluded unless directly relevant to architectural design methodologies.

Rather than pursuing exhaustive bibliometric coverage, the review adopted a theoretically oriented interpretive strategy aimed at identifying the major computational logics, epistemological transformations, and infrastructural relationships shaping the evolution of architectural computation. The methodology combines three main components: (i) diachronic mapping of computational paradigms; (ii) comparative analysis; and (iii) theoretical categorization. Accordingly, the method operates at three interconnected levels, a diachronic mapping that situates technologies and practices within their technical and cultural contexts; a transversal comparative analysis applied consistently across digital paradigms; and a theoretical categorization that organizes the findings into interpretive models. Rather than evaluating the technical performance of digital tools, the study seeks to understand how different computational regimes coexist, interact, and reconfigure processes of conception, representation, automation, control, and decision-making in architectural design [4,12].

The temporal scope extends from the first human–machine interactions in architecture in the 1950s and 1960s [25] to contemporary approaches based on machine learning and artificial intelligence (AI) [8,9]. In a first stage, the material is organized through a diachronic framework combining temporal periods and analytically distinct computational paradigms. While some paradigms overlap chronologically, they are differentiated according to their epistemological logic, operational structure, and modes of computational interaction.

The analysis identifies five major computational paradigms:

- (1) foundations of computation and early applications in architecture (1950–1970);
- (2) consolidation of CAD and the emergence of informational models (1970–1990);
- (3) dissemination of BIM (1990–2010);
- (4) development of parametric and algorithmic design (1990–2010);
- (5) integration of AI, machine learning, and data-driven computational systems (2010–2025).

Although BIM and parametric design overlap temporally within the 1990–2010 period, they are treated as analytically distinct paradigms rather than as a single chronological phase. BIM consolidates an object-oriented informational logic centered on coordination, interoperability, and lifecycle integration, whereas parametric and algorithmic design operate through rule-based generative systems focused on formal variation and performance-driven exploration. Their differentiation is therefore epistemological and operational rather than strictly chronological.

Vertically, the matrix is structured around analytical axes capturing epistemological, operational, and disciplinary implications, including:

- Nature of computation
- Type of digital model
- Human–machine relationship
- Degree of automation
- Learning/adaptivity
- Role of data
- Authorship and agency
- Contribution to the design process
- Impact on professional practice
- Risks and limitations

Rather than functioning as a chronological taxonomy of digital tools, the matrix is conceived as a comparative interpretive framework for analysing the coexistence, operational logic, and epistemological implications of computational paradigms in architecture.

The matrix is read both diachronically and transversally. Diachronic reading allows the identification of long-term structural transformations in architectural computation, while transversal reading enables the comparison of paradigms across specific analytical criteria. By explicitly defining comparative criteria and interpretive logic, this framework grounds the analysis developed in subsequent sections and is synthesised in Table 1. The overall methodological structure adopted in this study is illustrated in Figure 1.

Table 1. Comparative Analytical Matrix.

Analytical Axis	Foundations of Computation & Early Applications (1950–1970)	Consolidation of CAD & Emergence of Informational Models (1970–1990)	Dissemination of BIM (1990–2010)	Parametric & Algorithmic Design (1990–2010)	AI Integration & Data-Driven Computational Systems (2010–2025)
Nature of computation	Symbolic, logical, deterministic	Graphical, procedural	Rule-based, object-oriented	Algorithmic, rule-based	Data-driven, probabilistic
Type of digital model	Abstract, experimental	Geometric (2D/early 3D)	Informational model with parametric objects	Generative parametric systems	Trained models (ML/DL, generative)
Human–machine relationship	Exploratory, research-oriented	Operator–tool	Supervision and coordination	Designer as rule author	Curation, supervision

Table 1. Cont.

Analytical Axis	Foundations of Computation & Early Applications (1950–1970)	Consolidation of CAD & Emergence of Informational Models (1970–1990)	Dissemination of BIM (1990–2010)	Parametric & Algorithmic Design (1990–2010)	AI Integration & Data-Driven Computational Systems (2010–2025)
Degree of automation	None or minimal	Low (drafting efficiency)	Medium (coordination, verification)	Medium–High (controlled generation)	High (sometimes near total)
Learning/adaptivity	None	None	None	None (explicit rules only)	Present (machine learning)
Role of data	Negligible, unstructured	Implicit, non-semantic	Central, structured, interoperable	Limited (parameters, not datasets)	Foundational (datasets, bias, provenance)
Authorship and agency	Fully human	Fully human	Human, distributed	Hybrid (human + algorithm)	Deeply hybrid, contested
Contribution to design process	Conceptual foundations	Efficiency, precision, standardisation	Coordination, lifecycle integration	Formal exploration, performance optimisation	Generation, optimisation, prediction
Impact on professional practice	Marginal (academic)	High (Practice-wide adoption)	Very high (AEC transformation)	High (advanced practices, academia)	Emerging, potentially disruptive
Risks and limitations	Technological immaturity	Illusion of intelligence	Standardisation, rigidity	Over-formalism, opacity	Opacity, bias, IP, loss of control

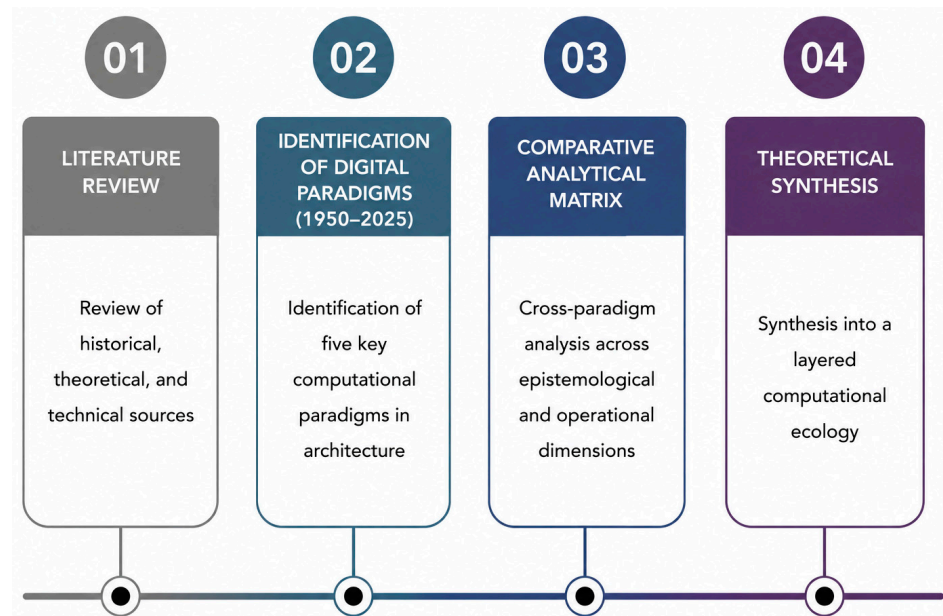


Figure 1. Methodological framework. The study adopts a four-stage methodological structure combining historical-critical review, diachronic identification of computational paradigms, comparative analytical interpretation, and interpretive synthesis. The framework integrates literature review, cross-paradigm analysis, and epistemological categorisation in order to interpret the long-term evolution of architectural computation as a layered computational ecology.

Table 1 operationalises the comparative framework by mapping the five temporal paradigms against key epistemological, operational, and disciplinary dimensions, enabling both diachronic and transversal interpretation. The matrix is intended to reveal both continuities and transformations across paradigms, emphasising how distinct computational logics remain active and interconnected within contemporary architectural workflows. Each analytical axis presented in the matrix is further examined and interpreted throughout Sections 5 and 6, allowing the table to function not merely as a classificatory summary but as a synthetic representation of the comparative arguments developed in the article.

4. Computational Paradigms in Architecture: A Diachronic Framework

4.1. Digital Paradigms as Coexisting Computational Regimes

The concept of “computational paradigm” adopted in this study does not imply a strictly linear technological progression, nor a deterministic succession of tools replacing one another. Rather, it refers to historically situated but epistemologically structured regimes of computation that configure distinct modes of thinking, designing, modelling, and interacting with digital systems [30–32].

In this sense, the notion of paradigm resonates partially with Kuhn’s understanding of scientific paradigms as coherent constellations of practices, assumptions, and problem-solving frameworks [33]. However, architectural computation does not evolve through abrupt paradigm shifts in the strict Kuhnian sense. Instead, it operates through layered coexistence, hybridization, and gradual reconfiguration. Earlier regimes persist within contemporary practice, even as new computational logics emerge.

For instance, rule-based drafting systems developed during the consolidation of CAD remain operative within BIM environments; parametric logics coexist with object-oriented informational models; and algorithmic scripting continues to function alongside data-driven AI systems [12,13]. The discipline therefore operates within a stratified computational ecology rather than a linear technological timeline.

Understanding digital paradigms as coexisting computational regimes allows us to interpret architectural computation not as a unidirectional narrative of progress, but as a dynamic field in which symbolic, procedural, parametric, informational, and probabilistic systems overlap and interact. This perspective prevents reductive technological determinism and foregrounds the cultural, disciplinary, and epistemological implications of computational integration in architecture [11,16].

4.2. Analytical Axes: Automation, Agency, Authorship, Data, Control

To analyze these regimes comparatively, it is necessary to move beyond descriptive accounts of software evolution and instead focus on structural dimensions that redefine architectural practice. The analytical axes adopted in this study—automation, agency, authorship, data, and control—capture these transformations at an epistemological and operational level [10,11].

Automation refers to the degree to which computational systems execute predefined operations, generate formal variations, coordinate information, or produce outputs autonomously. In early symbolic regimes (1950–1970), automation was minimal and primarily calculative. With CAD, automation enhanced drafting efficiency. BIM introduced coordinated object-based automation. Parametric systems enabled controlled generative automation. Contemporary AI systems, by contrast, incorporate probabilistic and learning-based automation capable of producing outputs not explicitly predefined by the designer [6,15,25].

Agency concerns the distribution of decision-making between human and machine. In early computational experiments, the architect remained the sole cognitive agent. With parametric systems, agency became partially delegated to rule-based generative procedures.

In AI-driven environments, agency becomes hybrid and negotiated, raising questions about machine participation in design reasoning [12,34].

Authorship evolves accordingly. The transition from drawing as representation to modelling as rule-definition shifts the architect's role from form-maker to system-author [9]. In AI contexts, authorship becomes deeply hybrid, involving dataset curation, prompt engineering, model training, and output selection. This produces contested notions of intellectual ownership and creative responsibility [34,35].

Data plays a progressively central role across paradigms. In early computational regimes, data was negligible or unstructured. CAD treated geometry primarily as graphical information. BIM formalized structured object-based data and lifecycle integration [4]. Parametric systems operate on relational parameters but not necessarily large datasets. AI regimes, however, are fundamentally data-dependent, with performance conditioned by training corpora, bias structures, and model architectures [6,25].

Finally, control shifts from deterministic symbolic manipulation to increasingly probabilistic and opaque systems. Whereas early computational systems were fully transparent and rule-governed, contemporary AI models introduce stochastic processes and black-box dynamics that complicate predictability and accountability [36].

These axes do not merely describe technological attributes; they articulate transformations in epistemology, professional identity, and disciplinary culture. They therefore provide the conceptual scaffolding for interpreting the five temporal moments articulated below.

4.3. Overview of the Five Temporal Moments

Based on the diachronic framework introduced in Section 3, five major temporal configurations of architectural computation can be identified. These are not mutually exclusive stages but analytically distinct regimes that partially overlap and coexist:

1. Foundations of computation and early applications (1950–1970)—characterized by symbolic, logical, and experimental uses of computation, primarily within academic and research contexts. The computer functioned as a calculative and exploratory instrument rather than a professional drafting tool [13,24].
2. Consolidation of CAD and emergence of informational models (1970–1990)—marked by the diffusion of computer-aided drafting systems, increased professional adoption, and the procedural stabilization of digital drawing. During this period, geometry became programmable and standardized, but remained largely representational [4,10].
3. Dissemination of BIM (1990–2010)—defined by object-oriented modelling, interoperability, and lifecycle integration. Here, the digital model evolved into an informational infrastructure central to coordination and professional transformation across the AEC industry [7].
4. Development of parametric and algorithmic design (1990–2010)—temporally overlapping with BIM but conceptually distinct. Parametric systems foreground rule-based generative logic, formal exploration, and performance optimization, positioning the architect as author of relational systems rather than static forms [4–6].
5. Integration of AI and advanced automation debates (2010–2025)—characterized by machine learning, generative models, predictive analytics, and emerging discussions on advanced automation. In this regime, probabilistic reasoning, data dependency, and adaptive systems redefine the boundaries of design agency and automation [8,9,34].

The separation of these moments serves analytical clarity rather than chronological rigidity. Their coexistence reflects the layered structure of contemporary practice, in which CAD drawings, BIM coordination, parametric scripting, and AI-assisted generation may operate simultaneously within a single project environment.

The five temporal moments outlined above constitute the historical and conceptual structure of the study. Each is examined as a distinct yet coexisting computational regime, allowing for a deeper understanding of its internal logic, operative mechanisms, and disciplinary implications. The temporal overlap and coexistence of these computational paradigms are illustrated in Figure 2.

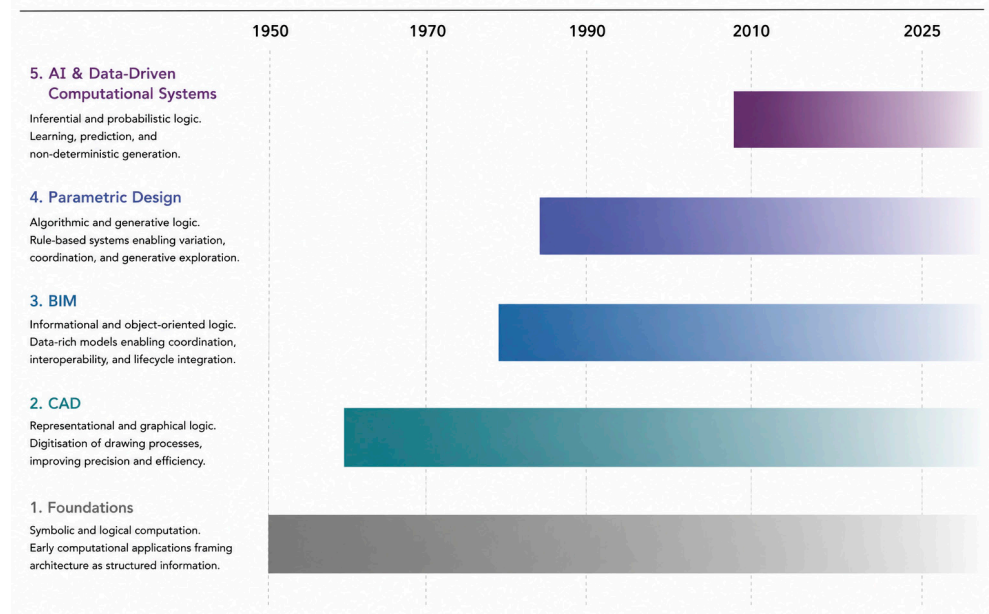


Figure 2. Diachronic overlap of computational paradigms in architecture (1950–2025). The diagram illustrates the temporal coexistence and partial overlap of five major computational paradigms in architecture: computational foundations, CAD, BIM, parametric design, and AI-based systems. Rather than representing a linear succession of technological replacements, the figure emphasises the persistence, interoperability, and simultaneous operation of multiple computational logics within contemporary architectural practice.

5. The Five Digital Paradigms of Architecture

5.1. Foundations of Computation and Early Applications in Architecture (1950–1970)

The emergence of computation in architecture between 1950 and 1970 must be situated within the broader post-war transformation of science, engineering, and systems theory. Rather than constituting an immediate digitalization of architectural practice, this period represents an epistemological shift in how architectural problems were conceptualized, structured, and processed [15,37].

5.1.1. From Numerical Calculation to Structured Problem-Solving

Early digital computers were primarily developed for military and scientific applications, including ballistics, cryptography, and structural engineering [23]. In architectural contexts, their first uses were linked to numerical analysis, structural calculations, optimization of load distribution, and environmental performance simulations [1].

Programming languages such as FORTRAN enabled engineers to encode mathematical procedures into executable sequences [38]. Although graphical interfaces were not yet available, the conceptual implications were significant: architectural problems could now be decomposed into variables, constraints, and iterative procedures. This alignment between architecture and formalized computation echoed developments in operations research and decision theory, particularly Herbert Simon's theory of bounded rationality [39].

Herbert A. Simon's notion of design as a problem-solving activity profoundly influenced early computational design thinking. In *The Sciences of the Artificial*, Simon

framed design as a search process within a structured problem space—an approach highly compatible with computational logic [39].

Thus, even before graphical CAD systems emerged, computation had begun to redefine architecture as a domain of structured information processing.

5.1.2. Cybernetics and Systems Theory

The theoretical foundations of early computational architecture were deeply informed by cybernetics and systems theory. Norbert Wiener introduced feedback, control, and adaptive behaviour as central concepts for understanding complex systems [40]. These ideas were rapidly absorbed into architectural discourse during the 1960s [41,42].

Architecture began to be interpreted as a dynamic system composed of interrelated subsystems—structural, environmental, and social—regulated through feedback loops. Gordon Pask’s work on conversational theory influenced experimental architectural thinking, notably in Cedric Price’s Fun Palace project [41,42].

Reyner Banham reframed buildings as environmental control systems rather than purely formal compositions [42]. This systems-based perspective laid the intellectual groundwork for later computational modelling and parametric dependency structures.

Cybernetics introduced a conceptual continuity that persists across digital paradigms, architecture could be modelled as a system of information flows rather than a static object.

5.1.3. The Design Methods Movement and Algorithmic Thinking

The 1960s also witnessed the rise of the Design Methods Movement, which sought to rationalize and systematize architectural design [43]. Christopher Alexander proposed decomposing design problems into structured diagrams of relationships and constraints [44]. Although not computational in execution, Alexander’s method exhibited algorithmic logic.

Similarly, J. Christopher Jones advocated structured design procedures based on analysis, generation, and evaluation cycles [45]. These approaches paralleled computational sequencing and anticipated rule-based workflows [1].

While critics later warned against excessive rationalization [46], the formalization of design logic created fertile ground for digital implementation [1].

Thus, computation during this period operated not merely as hardware but as an emerging epistemic model.

5.1.4. Early Interactive Computer Graphics

A decisive milestone occurred with Ivan Sutherland’s Sketchpad (1963), which introduced interactive graphical manipulation, constraint-based geometric relationships, object hierarchies, and parametric dependencies [25].

Sketchpad demonstrated that geometry could be defined relationally rather than statically. Constraints ensured that changes in one element propagated through the system, a principle that would later underpin parametric modelling and BIM environments [9].

Although hardware limitations prevented immediate architectural adoption, Sketchpad marked the transition from numerical to graphical computing and initiated a shift from representational drawing to computational modelling [5].

5.1.5. Experimental Architectural Cybernetics

Beyond analytical computation, late-1960s architectural experiments began to explore computation as an interactive and adaptive medium. Cedric Price’s Fun Palace (1961–1964), developed in collaboration with Gordon Pask, proposed an architecture capable of spatial reconfiguration through user participation and feedback mechanisms [47]. Although technologically unrealised, the project embodied a proto-computational logic

in which architecture was conceived as a dynamic system regulated by information flows rather than fixed geometry.

In parallel, Nicholas Negroponte's Architecture Machine Group at MIT investigated human–computer interaction in design processes [13]. Rather than treating the computer as a drafting instrument, Negroponte envisioned it as an intelligent assistant capable of engaging in a dialogue with the designer. This speculative shift—from tool to partner—prefigures later debates on AI-assisted design and machine agency.

These experiments introduced a crucial conceptual displacement, computation began to be understood not only as a mechanism for calculation or representation but as a cognitive interface capable of mediating decision-making processes.

5.1.6. Constraints and Institutional Limits

Despite profound conceptual advances, the practical adoption of computational technologies in architectural offices remained extremely limited. The high cost of mainframe systems, restricted institutional access, programming complexity, and rudimentary graphical output constrained experimentation to research laboratories and academic institutions [15].

As Picon observes, the gap between conceptual innovation and professional implementation persisted for decades, with computation functioning primarily as an intellectual horizon rather than a normalized technical instrument during this period [15].

5.1.7. Epistemic Foundations of Later Paradigms

Although the 1950–1970 period did not yet produce fully digital architecture in practical terms, it established the epistemic infrastructure upon which subsequent paradigms would be constructed:

- Architecture conceived as an information system
- Design understood as structured problem-solving
- Geometry interpreted as relational and rule-governed
- Feedback recognised as a regulatory mechanism

These conceptual shifts constituted a proto-paradigm, a cognitive transformation preceding technical normalization. As Carpo argues, digital architecture did not begin with software adoption but with the gradual reconceptualization of architecture as code, data, and system [6].

In this sense, the foundational era represents not a technological revolution in built output, but a paradigmatic reorientation in architectural knowledge—one that redefined how architectural problems could be structured and processed [15].

5.2. Consolidation of CAD and the Emergence of Informational Models (1970–1990)

The period between 1970 and 1990 marks the transition from experimental and conceptual uses of computation to its gradual integration into professional architectural practice. This phase is characterized by the consolidation of Computer-Aided Design (CAD) as a representational and drafting tool, enabling the digitization of drawing processes and the standardization of graphic production [1,23].

5.2.1. From Experimental Computing to Professional Tool

Advances in hardware, including the development of minicomputers and later personal computers, significantly expanded access to computational technologies beyond research institutions [23]. At the same time, the emergence of interactive graphic systems and commercial software made digital tools increasingly viable within architectural offices.

Early CAD systems such as Intergraph and AutoCAD (Autodesk, Inc., San Francisco, CA, USA), released in 1982, introduced digital drafting environments that replicated and

enhanced traditional drawing practices. These systems allowed for increased precision, editing flexibility, and reproducibility, reducing the time required for documentation and facilitating iterative design processes [1].

Despite these advances, CAD remained fundamentally representational, geometry was encoded digitally but retained the logic of conventional drawing. The computer functioned primarily as an instrument for visualization and documentation rather than as an active generator of form.

5.2.2. Procedural Logic and Early Computational Workflows

Although CAD systems were largely used for drafting, they also introduced procedural logic into design workflows. Layers, blocks, and coordinate systems enabled a structured organization of geometric information, anticipating later developments in object-oriented modelling.

This period also witnessed early experiments in algorithmic design and shape generation, particularly in academic contexts. However, these remained marginal to mainstream professional practice, which was primarily focused on efficiency and standardization [43].

Thus, while computation began to influence architectural production processes, its epistemological impact remained limited compared to the conceptual transformations initiated in the previous period.

5.2.3. Standardization and Professionalization

One of the most significant contributions of CAD was the standardization of architectural representation. Digital drafting facilitated consistency across projects, improved coordination between disciplines, and supported the emergence of shared conventions in drawing production.

This standardization aligned with broader transformations in the architecture, engineering, and construction (AEC) industry, where increasing project complexity demanded more efficient documentation and communication systems [1].

However, this efficiency-oriented paradigm also reinforced a separation between design thinking and representation. The computer was predominantly used to execute predefined decisions rather than to actively participate in design generation.

5.2.4. Limits of Representational Computation

Despite its widespread adoption, CAD did not fundamentally alter the underlying logic of architectural design. The absence of embedded data, relational dependencies, and integrated performance feedback limited its capacity to operate as a true computational design environment.

As a result, CAD can be understood as a transitional paradigm, it digitalized representation without fully transforming the epistemological structure of design processes. This limitation would later motivate the development of BIM and parametric systems, which sought to overcome the separation between geometry, information, and process [4,12].

5.2.5. Transitional Role in the Computational Ecology

Within the broader diachronic framework proposed in this study, the CAD paradigm occupies a transitional position. It establishes the technical and professional infrastructure necessary for subsequent developments while maintaining strong continuity with pre-digital representational practices.

Rather than constituting a radical epistemological rupture, CAD can be understood as a phase of translation, the migration of analogue drawing into a digital environment. Its primary contribution lies in the digitization and standardization of representation, rather than in the transformation of design logic itself.

This transitional condition explains both its widespread adoption and its limitations. While CAD significantly increased efficiency and precision, it did not yet integrate information, relational dependencies, or generative processes. These limitations would later motivate the emergence of BIM and parametric systems as more advanced computational regimes [4,12].

The legacy of CAD persists in contemporary practice, where drafting-based workflows continue to coexist with information-based and generative paradigms, reinforcing the layered and non-linear nature of architectural computation.

5.3. Dissemination of BIM (1990–2010)

The period between 1990 and 2010 marks a fundamental transformation in architectural computation, characterized by the transition from representational systems to information-based modelling environments. Building Information Modelling (BIM) introduced a new paradigm in which geometry is no longer treated as isolated graphical entities but as part of an integrated system of data-rich objects [7].

5.3.1. From Representation to Information

Unlike CAD, where drawings encode geometry without embedded meaning, BIM systems structure architectural elements as parametric objects containing both geometric and semantic information. Walls, slabs, and components are no longer lines or surfaces but entities defined by attributes, relationships, and behaviours.

This shift enables the model to function as a shared informational database rather than a static representation. As Eastman et al. argue, BIM integrates design, construction, and lifecycle management into a unified digital environment, fundamentally altering workflows across the AEC industry [7].

5.3.2. Object-Oriented Modelling and Interoperability

BIM is grounded in object-oriented logic, where elements are defined through classes, properties, and relational dependencies. This allows for coordinated updates, consistency across views, and the automation of documentation processes.

Interoperability becomes a central concern in this paradigm. Standards such as IFC (Industry Foundation Classes) facilitate data exchange between different software platforms, enabling multidisciplinary collaboration and integrated project delivery [7].

This capacity for coordination distinguishes BIM from earlier computational systems and positions it as a central infrastructure in contemporary architectural practice.

5.3.3. Integration and Lifecycle Thinking

A key innovation of BIM lies in its ability to integrate multiple phases of the building lifecycle within a single model. Design, analysis, construction, and facility management are interconnected through shared data structures.

This lifecycle-oriented approach reflects broader shifts toward systems integration and information management in the AEC industry. It also aligns with emerging demands for sustainability, performance analysis, and long-term asset management.

However, this integration is not purely technical; it also entails organizational and professional transformations, redefining roles, responsibilities, and workflows within architectural practice [1].

5.3.4. Limits of Informational Modelling

Despite its transformative impact, BIM remains constrained by its underlying logic. While it integrates information and coordination, it does not inherently generate form or enable exploratory design processes.

The model is typically defined through predefined object libraries and standardized components, which may limit formal experimentation and creative variation. As a result, BIM can be seen as a system optimized for coordination and control rather than for generative design.

This limitation explains the parallel emergence of parametric and algorithmic design systems, which address precisely the absence of generative capacity within BIM environments [4,5].

5.3.5. BIM Within the Computational Ecology

Within the diachronic framework proposed in this study, BIM represents a consolidation of information-based computation in architecture. It establishes a stable and widely adopted paradigm centered on coordination, interoperability, and lifecycle integration.

However, rather than replacing previous systems, BIM coexists with CAD-based workflows and overlaps with parametric design practices. This coexistence reinforces the notion of a layered computational ecology, in which multiple paradigms operate simultaneously within architectural production [12].

Thus, BIM should be understood not as an endpoint but as a central layer within a broader and evolving computational landscape.

5.4. Development of Parametric and Algorithmic Design (1990–2010)

Parallel to the dissemination of BIM, the period between 1990 and 2010 witnessed the emergence of parametric and algorithmic design as a distinct computational paradigm. While BIM focused on information integration and coordination, parametric systems introduced rule-based generative processes that fundamentally transformed the relationship between geometry, logic, and design [5,6].

5.4.1. From Static Geometry to Relational Modelling

Parametric design is based on the definition of relationships between elements rather than on fixed geometric entities. Instead of modelling forms directly, architects define parameters, constraints, and dependencies that govern the behaviour of the system.

This relational approach allows for continuous variation, adaptation, and iteration. Changes in one parameter propagate throughout the system, generating families of forms rather than singular solutions. As Schumacher argues, parametricism represents a shift toward a logic of differentiation, correlation, and continuous transformation [5].

5.4.2. Algorithmic Thinking and Generative Processes

Closely linked to parametric modelling, algorithmic design introduces scripting and computational procedures as central design tools. Architects increasingly operate through sequences of instructions, defining rules that generate geometry rather than manually constructing it.

This shift aligns architectural design with computational logic, where form emerges from processes rather than being predefined. Terzidis describes this transition as a move from composition to computation, emphasizing the role of algorithms as drivers of design thinking [4].

Algorithmic approaches enable not only formal exploration but also performance-driven design, where environmental, structural, or material criteria inform the generation of form [6].

5.4.3. Material Computation and Performance

A key development within this paradigm is the integration of material behaviour and performance criteria into the design process. Menges argues that computation allows

for a higher level of integration between material properties, fabrication processes, and geometric definition [6].

This approach shifts design from representation toward simulation and optimization, where digital models operate as predictive systems capable of evaluating multiple design scenarios.

5.4.4. Expanded Agency and Control

Parametric and algorithmic systems redistribute agency between the architect and the computational model. While the designer defines rules and parameters, the system generates multiple possible outcomes, introducing a controlled form of automation.

This hybrid condition transforms the architect's role into that of a system designer or process author. However, unlike AI systems, parametric models remain fully deterministic and transparent, allowing for complete control and traceability of design decisions [4].

5.4.5. Limits and Critiques

Despite its generative capacity, parametric design has been subject to criticism. Some authors argue that it risks privileging formal complexity over social, cultural, or contextual considerations.

Furthermore, the emphasis on continuous variation and differentiation may lead to stylistic homogenization, as similar computational logics produce comparable formal outcomes across different contexts [12].

These critiques highlight the need to situate parametric design within broader disciplinary frameworks rather than treating it as a purely technical advancement.

5.4.6. Parametric Design Within the Computational Ecology

Within the diachronic framework proposed in this study, parametric and algorithmic design represent a paradigm centered on generative logic, relational modelling, and performance-driven processes.

Rather than replacing BIM, parametric systems often operate in parallel or in integration with informational models, contributing to a multi-layered computational environment. This coexistence reinforces the understanding of architectural computation as a stratified ecology of interacting paradigms [12].

5.5. *Integration of AI and Data-Driven Computational Paradigms (2010–2025)*

The most recent phase in the evolution of computational paradigms in architecture is characterized by the integration of artificial intelligence (AI), machine learning (ML), and data-driven design processes. Unlike previous paradigms based on deterministic and rule-based systems, AI introduces probabilistic models capable of learning patterns from data and generating outputs that are not explicitly predefined [8].

This shift represents a significant epistemological transformation. Computation is no longer limited to executing instructions but increasingly involves inference, prediction, and adaptive behaviour. As a result, architectural design processes are progressively influenced by systems that operate through statistical reasoning and pattern recognition rather than explicit rule definition [8,9].

AI applications in architecture currently operate across multiple domains, including generative design, spatial layout optimization, image-based conceptualization, and performance prediction. These systems rely on large datasets, neural network architectures, and training processes that fundamentally reshape the relationship between data, model, and output [8,25].

5.5.1. Generative Spatial Production and Machine Learning Models

One of the most significant developments within this paradigm is the emergence of generative models capable of producing architectural layouts, images, and spatial configurations. Generative Adversarial Networks (GANs), diffusion models, and transformer-based architectures have enabled new forms of design exploration based on data-driven synthesis rather than explicit modelling procedures [8,48].

Research such as ArchiGAN demonstrates how trained neural networks can generate floor plans from spatial requirements using conditional generative frameworks. These systems operate by learning statistical correlations between programmatic inputs and spatial configurations, effectively translating architectural knowledge into latent representations [48].

Recent developments in large language models and generative AI systems have further expanded the role of artificial intelligence within architectural workflows. Beyond image generation, contemporary AI systems are increasingly integrated into conceptual ideation, early-stage design exploration, spatial organisation, programmatic analysis, and human–AI collaborative design processes. Emerging research on co-design environments suggests that AI is progressively functioning not only as an optimisation tool, but as an interactive computational collaborator capable of participating in iterative and conversational design exploration [9,49–52].

At the same time, these developments intensify concerns regarding authorship, epistemic opacity, dataset bias, and the homogenisation of spatial solutions generated through recurrent dataset logics and statistically dominant architectural patterns [49–51].

Beyond experimental research environments, AI-driven generative approaches are increasingly being integrated into applied architectural workflows. For example, recent studies on AI-assisted layout generation and urban-scale design optimisation have demonstrated how machine learning systems can generate spatial configurations based on environmental, regulatory, and programmatic constraints, producing multiple design alternatives for evaluation [50,52].

Platforms such as Spacemaker (now integrated into Autodesk Forma, Autodesk, Inc., San Francisco, CA, USA) exemplify this transition at the urban and architectural scale, where generative algorithms and machine learning techniques are used to optimise building massing, daylight access, noise exposure, and spatial distribution in early design stages. In these contexts, designers engage with probabilistic outputs rather than predefined geometries, evaluating and refining machine-generated scenarios [51,52].

This shift illustrates a fundamental transition from deterministic modelling to probabilistic exploration, reinforcing the notion that AI operates not only as a representational tool but as an active computational agent within the design process [5,6,50].

Similarly, recent diffusion-based models and AI-assisted platforms have demonstrated the capacity to generate architectural imagery and conceptual proposals from textual or visual prompts. While some of these tools remain experimental or commercially oriented, they illustrate the rapid expansion of generative capabilities within architectural workflows [49,51].

As Leach argues, AI-driven design introduces a new paradigm in which architectural production is increasingly mediated by datasets, training processes, and algorithmic inference, rather than by direct geometric control [9].

5.5.2. Data Dependency, Bias, and Epistemological Implications

A defining characteristic of AI-based systems is their dependence on data. Unlike parametric or algorithmic models, which operate through explicitly defined rules, AI

systems derive their behaviour from training datasets. This introduces new challenges related to data quality, representativeness, and bias [8,49].

Bias embedded in training data can lead to systematic distortions in generated outputs, raising critical ethical and disciplinary questions. In architectural contexts, this may affect spatial typologies, cultural representation, and design assumptions embedded within generative systems [49].

Furthermore, the opacity of many machine learning models—often described as “black-box” systems—complicates issues of transparency and control. Designers may not fully understand how specific outputs are generated, challenging traditional notions of authorship, responsibility, and decision-making [8].

These conditions mark a fundamental shift in the epistemology of architectural computation: from deterministic and interpretable systems toward probabilistic, data-driven, and partially opaque processes.

5.5.3. Hybrid Agency and the Transformation of Authorship

The integration of AI into architectural workflows produces a hybrid model of agency in which human and machine contributions are deeply intertwined. Designers increasingly operate through processes of selection, curation, and evaluation of machine-generated outputs rather than direct form creation.

This transformation extends and intensifies tendencies already present in parametric design, but with a crucial difference: AI systems introduce elements of unpredictability and autonomy that exceed rule-based generative logic [12].

As a consequence, authorship becomes distributed across multiple actors, including the designer, the dataset, and the computational model. This raises complex questions regarding intellectual ownership, creative responsibility, and the role of the architect within AI-mediated design processes [9,49–51].

5.5.4. Toward a Probabilistic Paradigm of Architectural Computation

Within the diachronic framework proposed in this study, AI represents a distinct computational paradigm characterized by probabilistic reasoning, data dependency, and adaptive behaviour. Rather than replacing previous paradigms, AI systems coexist with CAD, BIM, and parametric models, contributing to an increasingly complex and layered computational ecology.

This paradigm extends the trajectory of architectural computation toward systems capable not only of modelling and generating form, but also of learning, predicting, and interacting with uncertain and dynamic conditions. In this sense, AI does not simply introduce new tools, but fundamentally redefines the cognitive and operational foundations of architectural design.

The transition from deterministic computational paradigms to AI-based learning systems introduces a fundamental transformation in how architectural design processes are structured, controlled, and interpreted. While previous paradigms—such as CAD, BIM, and parametric design—operate through explicit rules and fully traceable procedures, AI systems rely on probabilistic inference, data-driven training, and partially opaque models.

To synthesize these differences, Table 2 presents a comparative framework contrasting deterministic and learning-based computational regimes across key analytical dimensions introduced in Section 4.2.

Table 2. Comparative framework contrasting deterministic computational systems (CAD, BIM, parametric design) and AI-based learning systems across key analytical dimensions, including rule transparency, causal traceability, adaptivity, data dependence, and agency distribution.

Analytical Dimension	Deterministic Systems (CAD/BIM/Parametric)	Learning Systems (AI)
Rule transparency	Explicit, rule-based, fully defined	Implicit, model-based, partially opaque
Causal traceability	High; processes are traceable and reproducible	Limited; outputs emerge from trained models
Adaptivity	Fixed unless manually redefined	Dynamic; capable of learning and adaptation
Data dependence	Low to moderate; parameter-driven	High; performance depends on training data
Agency distribution	Primarily human-controlled	Hybrid; shared between human and machine

This comparison highlights a critical shift from transparent and deterministic systems toward adaptive and data-dependent computational models. Rather than representing a simple technological progression, this transformation reconfigures the epistemological foundations of architectural design. Control becomes distributed, authorship hybridized, and decision-making processes increasingly mediated by statistical inference.

Importantly, this distinction does not imply the replacement of deterministic paradigms by AI systems. Instead, it reinforces the notion of a layered computational ecology in which multiple regimes coexist, interact, and operate simultaneously within contemporary architectural practice. This coexistence provides the conceptual basis for the comparative analysis developed in the following section.

6. Comparative Analysis

The analysis developed in this section builds upon the analytical matrix and diachronic framework established in the methodology. By applying a consistent set of analytical axes across the identified computational paradigms, the study examines structural continuities, transformations, and interdependencies within architectural computation. The following subsections develop the principal analytical dimensions synthesised in Table 1, particularly the transformation of control, agency, data structures, adaptivity, and computational interdependence across paradigms.

Table 2 provides a synthetic comparison between deterministic systems—such as CAD, BIM, and parametric design—and AI-based learning systems. This comparison is not intended to establish a binary opposition, but to reveal fundamental shifts in how computation operates within architectural design processes.

6.1. Transformation of Control and Transparency

One of the most significant transformations concerns the nature of control. In deterministic systems (as synthesised in Table 1 under “nature of computation” and “logic of control”), control is explicit, rule-based, and fully traceable. Design processes can be reconstructed step by step, and outcomes are directly linked to predefined instructions. By contrast, AI-based systems introduce probabilistic inference, in which outputs are generated through learned patterns rather than explicit rules.

This transformation reduces transparency and introduces epistemic opacity. While parametric models remain interpretable, AI systems often operate as black-box processes, complicating the verification and validation of design decisions. As a result, control becomes indirect and mediated, relying on training data and model behaviour rather than explicit authorial intent. This shift complicates the long-standing assumption that

architectural design can be fully controlled through explicit rule-based systems, introducing forms of partial autonomy that redefine the scope of design responsibility.

6.2. Reconfiguration of Agency and Authorship

The redistribution of control is closely linked to changes in agency. In earlier paradigms, the architect retains primary authorship, even when working through parametric or algorithmic systems. In AI-based workflows, however, agency becomes hybrid and distributed, as reflected in the “authorship and agency” axis of Table 1.

Design outcomes are co-produced by the designer, the dataset, and the computational model. The architect’s role shifts from direct form generation toward processes of curation, selection, and evaluation. This transformation challenges traditional notions of authorship, raising questions about responsibility, ownership, and creative attribution.

Importantly, this condition does not eliminate human agency but redefines it within a broader socio-technical system. Rather than a loss of authorship, this transformation suggests a redistribution of creative responsibility across human and computational actors, requiring new frameworks for understanding design intention and accountability.

6.3. Data as a Design Driver

Another key distinction lies in the role of data. Deterministic systems operate primarily through parameters, rules, and structured inputs. While BIM introduces data-rich environments, this data remains explicitly defined and controlled.

In contrast, AI systems are fundamentally data-dependent, as outlined in the “role of data” dimension in Table 1. Their performance and outputs are conditioned by the quality, structure, and biases of training datasets. Data thus becomes not merely a support for design but a primary driver of form generation and decision-making.

This transformation introduces new epistemological and ethical challenges, including issues of bias, representational limits, and the implicit encoding of cultural assumptions within datasets. Consequently, data can no longer be assumed to function as neutral input, but as an active agent shaping design outcomes and disciplinary knowledge.

6.4. Adaptivity and Open-Ended Processes

Deterministic systems are inherently stable and predictable. Adaptation requires explicit modification of parameters or rules. AI systems, however, enable adaptive and open-ended processes, in which models can evolve through training, feedback, or iterative interaction.

This capacity for adaptivity expands the design space but also introduces uncertainty.

6.5. From Linear Evolution to Layered Computational Ecology

Taken together, these transformations demonstrate that computational paradigms in architecture do not evolve through simple technological substitution. Contemporary practice instead operates through the interaction of multiple computational logics, combining representational systems, informational models, generative processes, and data-driven inference within increasingly hybrid workflows.

In practice, CAD drafting, BIM coordination, parametric scripting, and AI-assisted generation are frequently deployed simultaneously within the same project environment, each contributing distinct operative capacities and epistemological assumptions. This condition reflects not a unified technological paradigm, but a heterogeneous computational ecology shaped by coexistence, interdependence, and partial overlap between systems. These interactions are not merely conceptual but operational. AI-assisted systems frequently rely on structured BIM datasets, parametric environments generate geometries later coordinated through BIM platforms, and CAD-based drafting workflows continue to

support documentation and representational processes within hybrid digital environments. Contemporary architectural computation therefore depends on relations of data exchange, workflow interoperability, and infrastructural layering between paradigms rather than on isolated technological systems. The resulting layered computational ecology and the relationships between the different computational paradigms are synthesised in Figure 3.

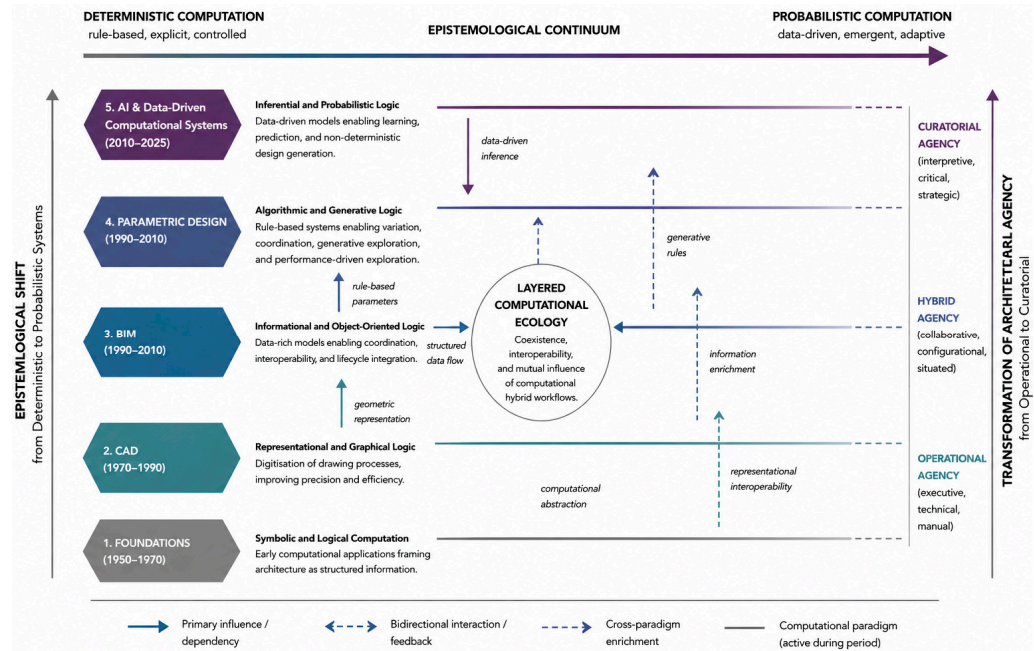


Figure 3. Layered computational ecology of architectural computation. The diagram illustrates the coexistence and interaction of five computational paradigms across an epistemological continuum extending from deterministic to probabilistic computation and through a parallel transformation of architectural agency from operational execution toward hybrid and curatorial forms of human–machine interaction. Rather than representing a linear technological succession, the framework emphasises layered interdependencies, interoperability, data exchange, and cross-paradigm enrichment between CAD, BIM, parametric systems, and AI-based computational environments operating simultaneously within contemporary architectural practice.

Rather than signalling the disappearance of previous paradigms, recent developments in AI reconfigure existing computational relationships while intensifying questions surrounding control, agency, authorship, and responsibility. In this sense, the comparative framework proposed in this study provides a structured basis for interpreting architectural computation beyond technology-centered narratives, positioning computation as a cultural, epistemological, and operational infrastructure within contemporary architectural practice.

7. Discussion

The comparative analysis developed in this study suggests that contemporary architectural computation is increasingly shaped by the interaction of heterogeneous computational logics rather than by discrete technological substitutions. This section discusses the implications of this condition through three interrelated dimensions: the transition from deterministic computation to adaptive intelligence, its consequences for architectural practice and education, and the ethical tensions associated with increasingly autonomous computational systems.

7.1. From Deterministic Computation to Adaptive Intelligence

A central finding of this research is that the most significant rupture in architectural computation does not lie in geometric complexity or representational sophistica-

tion, but in the transition from rule-based systems to learning-based inference. Earlier paradigms—CAD, BIM, and parametric modelling—operate within deterministic or explicitly scripted environments. Even when optimisation routines are involved, their logic remains traceable and human-authored [4,5,7].

Artificial intelligence introduces a qualitatively different computational logic. Machine learning systems derive internal representations from data rather than from predefined rules, enabling adaptive behaviour and probabilistic synthesis [6]. This transformation alters not only how architecture is produced, but how computational knowledge is constituted. Design decisions increasingly emerge from statistical correlations rather than explicit procedural logic.

This does not imply that architecture becomes autonomous or that designers relinquish control. Rather, agency is redistributed. Designers define datasets, curate outputs, and establish evaluation criteria, while learning systems explore solution spaces beyond explicitly scripted dependencies [9,12]. The shift is therefore conceptual and operational rather than merely technical. Architectural computation increasingly shifts from executing formal instructions toward negotiating probabilistic tendencies. Rather than indicating a loss of disciplinary control, this shift suggests a reconfiguration of architectural agency within a broader computational ecosystem.

Importantly, AI-based systems continue to rely on existing computational infrastructures, including BIM environments, parametric frameworks, and rule-based simulations [4,7]. Contemporary architectural computation therefore operates through the interaction of multiple digital frameworks rather than through the replacement of previous paradigms.

The long-term implication is that architectural computation increasingly resembles a hybrid computational environment in which symbolic logic, relational dependencies, and statistical inference operate simultaneously. Understanding this condition is essential for maintaining disciplinary coherence in the face of accelerating automation.

7.2. Implications for Architectural Practice and Education

The redistribution of computational agency has significant consequences for professional practice. As learning-based systems integrate into workflows, architects transition from direct form-givers to orchestrators of complex socio-technical systems. Design increasingly involves configuring constraints, curating data, and evaluating machine-generated alternatives.

These transformations require new competencies. Beyond geometric modelling or scripting proficiency, practitioners must develop literacy in data structures, model training processes, evaluation metrics, and infrastructural dependencies. Architectural expertise becomes partly infrastructural and epistemic: understanding how models are trained, how bias propagates, and how inference mechanisms operate [6].

Educational institutions therefore face a structural and curricular challenge. Traditional curricula organised around drawing, modelling, and representational skills must expand to include computational ethics, AI literacy, data governance, and critical algorithmic thinking. At the same time, foundational disciplinary knowledge—spatial reasoning, material logic, and environmental understanding—remains indispensable. The objective is not technological substitution but cognitive augmentation. Failing to address these transformations may result in a disciplinary gap between technological capability and critical architectural practice.

Moreover, the integration of AI demands renewed emphasis on critical reflection. Students must learn not only how to use generative systems but how to interrogate their assumptions, limitations, and socio-technical implications. Without such reflex-

ivity, architectural education risks producing operators of opaque systems rather than critical designers.

The profession is thus entering a phase in which design competence includes managing uncertainty, negotiating hybrid agency, and exercising judgement within probabilistic environments. Rather than diminishing architectural expertise, this evolution may redefine it as the capacity to synthesise computational outputs within broader cultural, spatial, and ethical frameworks.

These transformations also reflect the broader professional impacts synthesised in Table 1, including the transition from drafting efficiency toward increasingly adaptive and data-driven forms of architectural practice.

7.3. Risks, Limitations, and Ethical Considerations

While AI-driven systems expand generative capabilities, they also introduce a paradox within architectural production. The same systems that promise creativity and optimisation are fundamentally dependent on historical datasets, existing typologies, and inherited spatial conventions. As a result, generative AI may simultaneously increase formal variation while reinforcing dominant architectural norms embedded within training data. This tension suggests that technological innovation does not necessarily produce epistemological or cultural transformation.

The transition toward adaptive computational systems introduces structural risks that extend beyond technical performance. Three clusters of concern are particularly relevant.

First, opacity and explainability. Learning-based models often operate as black boxes, complicating traceability and accountability in design decisions [6]. In professional contexts involving safety, compliance, and public responsibility, opaque inference mechanisms raise governance challenges.

Second, bias and representational politics. AI systems trained on historical datasets may reproduce embedded social, cultural, or spatial inequalities [49]. Architectural outputs generated through such systems risk normalising existing biases in urban morphology, housing typologies, or environmental prioritisation.

Third, infrastructural dependence and control. AI tools increasingly rely on proprietary platforms, cloud-based infrastructures, and corporate ecosystems [11,49]. This dependency can shift power away from design actors toward technology providers, raising questions about intellectual autonomy, data ownership, and long-term professional sovereignty.

Additionally, adaptive systems are subject to instability, including model drift, distribution shifts, and unpredictable generalisation behaviours. In architecture—where performance failures have material consequences—such conditions cannot be treated as purely experimental issues.

This study also acknowledges its own limitations. The five-paradigm framework is interpretive and conceptual rather than empirical. It should therefore be understood as a heuristic analytical model rather than as a fixed historical taxonomy. While grounded in extensive literature review, it does not quantify adoption rates, productivity impacts, or performance metrics across paradigms. Future research may expand the comparative matrix through case studies, empirical analysis of workflows, or quantitative evaluation of AI integration. Future research may also investigate how AI-generated spatial systems influence architectural authorship, professional liability, aesthetic homogenisation, and decision-making processes in practice. Further empirical studies could examine how architects negotiate trust, control, and interpretability when interacting with probabilistic design systems.

Similarly, the risks synthesised in Table 1 vary across computational paradigms, ranging from technological immaturity and procedural rigidity in earlier systems to opacity, infrastructural dependency, and algorithmic bias in AI-based environments.

Ultimately, the integration of AI into architecture may represent not only a technological transition but a redefinition of how architectural knowledge itself is produced, validated, and exercised. The central challenge is therefore not whether AI will replace architectural judgement, but how the discipline will preserve critical agency within increasingly autonomous computational environments.

8. Conclusions

This study has demonstrated that the evolution of computational paradigms in architecture cannot be adequately understood as a linear sequence of technological innovations, but rather as a layered transformation in the epistemological and operational foundations of the discipline. Through the articulation of a diachronic framework and a comparative analytical matrix structured around five major paradigms—early computation, CAD, BIM, parametric design, and AI integration—the research examined how distinct computational regimes reconfigured the relationship between design, knowledge, and computation. Rather than rendering previous systems obsolete, these paradigms accumulate and interact, forming the layered computational ecology that characterises contemporary architectural practice.

The analysis developed through the five structural axes—automation, agency, authorship, data, and control—reveals that each paradigm introduces not only new technical instruments but also distinct modes of reasoning, representation, and professional agency. In particular, the transition from deterministic and rule-based systems toward probabilistic and data-driven models marks a significant epistemological transformation in architectural computation, redefining the relationship between human judgement and computational processes.

The integration of AI intensifies these transformations by introducing adaptive and data-driven forms of computation based on learned inference rather than explicit rule-following. This condition expands the design space while simultaneously raising concerns regarding opacity, bias, and the governance of computational decision-making within architectural practice.

A central contribution of this research is the notion of a layered computational ecology, which frames contemporary architectural practice not as dominated by a single paradigm but as constituted through the interaction of multiple computational logics linked through relations of interoperability, infrastructural dependency, and operational coexistence.

The implications of these findings extend to the redefinition of architectural expertise and professional identity. As computational systems become more adaptive and generative, the architect's role is increasingly defined by the capacity to exercise critical judgement across heterogeneous computational environments. This includes the ability to curate and evaluate data, to assess machine-generated outputs against spatial, cultural, and ethical criteria, and to maintain accountability in design decisions even when computational processes are not fully transparent. Rather than diminishing architectural knowledge, this transformation repositions it as a form of critical and ethical judgement within increasingly complex computational environments.

From this perspective, the ecological condition of architectural computation lies not only in the coexistence of paradigms, but in the continuous exchange of data, processes, and representational logics across interconnected computational systems.

In conclusion, architectural computation is best understood not as a succession of tools but as an evolving epistemological infrastructure that shapes how architecture represents,

generates, evaluates, and governs design processes. The framework of layered computational ecology proposed in this study provides a structured basis for critically interpreting this condition beyond technology-centred narratives. The future of architectural computation depends less on the dominance of a single paradigm than on the discipline's capacity to critically engage heterogeneous computational systems while maintaining responsibility for the spatial, cultural, and human dimensions of the built environment.

Author Contributions: Conceptualization, methodology, writing—original draft preparation, writing—review and editing: A.I.L.P., I.C.N.d.R.M. and R.J.G.R. All authors have read and agreed to the published version of the manuscript.

Funding: This work is financed by national funds through FCT—Fundação para a Ciência e a Tecnologia, I.P., under the Strategic Project with the references UID/04008: Centro de Investigação em Arquitetura, Urbanismo e Design.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: No new data were created or analyzed in this study.

Acknowledgments: The authors acknowledge the Directorate of the DAMG, UPT. The visualisations presented in this article were developed using generative AI-assisted workflows and graphic editing tools, including OpenAI ChatGPT, version GPT-5 (OpenAI, San Francisco, CA, USA), and Adobe Illustrator, version 2025 (Adobe Inc., San Jose, CA, USA). All analytical content, conceptual interpretation, and final figure editing were critically supervised and validated by the authors.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kolarevic, B. *Architecture in the Digital Age: Design and Manufacturing*; Taylor & Francis: New York, NY, USA, 2003.
2. Oxman, R. Theory and design in the first digital age. *Des. Stud.* **2006**, *27*, 229–265. [CrossRef]
3. Carpo, M. *The Second Digital Turn: Design Beyond Intelligence*; MIT Press: Cambridge, MA, USA, 2017.
4. Terzidis, K. *Algorithmic Architecture*; Architectural Press: Oxford, UK, 2006.
5. Schumacher, P. Parametricism: A new global style for architecture and urban design. *Archit. Des.* **2009**, *79*, 14–23. [CrossRef]
6. Menges, A. Material computation: Higher integration in morphogenetic design. *Archit. Des.* **2012**, *82*, 14–21. [CrossRef]
7. Eastman, C.; Teicholz, P.; Sacks, R.; Liston, K. *BIM Handbook: A Guide to Building Information Modeling for Owners, Managers, Designers, Engineers and Contractors*, 2nd ed.; Wiley: Hoboken, NJ, USA, 2011.
8. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016.
9. Leach, N. *Architecture in the Age of Artificial Intelligence: An Introduction to AI for Architects*; Bloomsbury: London, UK, 2021.
10. Mitchell, W.J. *The Logic of Architecture: Design, Computation, and Cognition*; MIT Press: Cambridge, MA, USA, 1990.
11. Picon, A. *Digital Culture in Architecture*; Birkhäuser: Basel, Switzerland, 2010.
12. Carpo, M. *The Alphabet and the Algorithm*; MIT Press: Cambridge, MA, USA, 2011.
13. Negroponte, N. *The Architecture Machine*; MIT Press: Cambridge, MA, USA, 1970.
14. Boole, G. *An Investigation of the Laws of Thought on Which are Founded the Mathematical Theories of Logic and Probabilities*; Walton and Maberly: London, UK, 1854.
15. Shannon, C.E. A symbolic analysis of relay and switching circuits. *Trans. Am. Inst. Electr. Eng.* **1938**, *57*, 713–723. [CrossRef]
16. Ifrah, G. *The Universal History of Computing: From the Abacus to the Quantum Computer*; Wiley: Hoboken, NJ, USA, 2001.
17. Swade, D. *The Difference Engine: Charles Babbage and the Quest to Build the First Computer*; Penguin Books: London, UK, 2000.
18. Turing, A.M. On computable numbers, with an application to the Entscheidungsproblem. *Proc. Lond. Math. Soc.* **1936**, *42*, 230–265. [CrossRef]
19. Copeland, B.J. *The Essential Turing: Seminal Writings in Computing, Logic, Philosophy, Artificial Intelligence, and Artificial Life*; Oxford University Press: Oxford, UK, 2004.
20. Turing, A.M. Computing machinery and intelligence. *Mind* **1950**, *59*, 433–460. [CrossRef]
21. Russell, S.; Norvig, P. *Artificial Intelligence: A Modern Approach*, 3rd ed.; Prentice Hall: Upper Saddle River, NJ, USA, 2010.
22. Goldstine, H.H. *The Computer from Pascal to von Neumann*; Princeton University Press: Princeton, NJ, USA, 1972.
23. Ceruzzi, P.E. *A History of Modern Computing*, 2nd ed.; MIT Press: Cambridge, MA, USA, 2003.

24. Steadman, P. Architectural design and computers. In *Architecture and the Computer Conference Proceedings*; Boston Architectural Center: Boston, MA, USA, 1965.
25. Sutherland, I.E. *Sketchpad: A Man-Machine Graphical Communication System*; Massachusetts Institute of Technology: Cambridge, MA, USA, 1963.
26. Otto, F.; Rasch, B. *Finding Form: Towards an Architecture of the Minimal*; Axel Menges: Stuttgart, Germany, 1996.
27. Mitchell, W.J. *City of Bits: Space, Place, and the Infobahn*; MIT Press: Cambridge, MA, USA, 1995.
28. Groat, L.; Wang, D. *Architectural Research Methods*, 2nd ed.; Wiley: Hoboken, NJ, USA, 2013.
29. Booth, A.; Sutton, A.; Papaioannou, D. *Systematic Approaches to a Successful Literature Review*; Sage: London, UK, 2016.
30. Vrachliotis, G. Computation, architecture and the production of knowledge. *Archit. Theory Rev.* **2013**, *18*, 300–315.
31. Leach, N. Digital morphogenesis. *Archit. Des.* **2009**, *79*, 32–37.
32. Picon, A. Architecture and the Sciences: Exchanging Metaphors. *Grey Room* **2003**, *12*, 78–97.
33. Kuhn, T.S. *The Structure of Scientific Revolutions*, 2nd ed.; University of Chicago Press: Chicago, IL, USA, 1970.
34. Floridi, L.; Chiriatti, M. GPT-3: Its Nature, Scope, Limits, and Consequences. *Minds Mach.* **2020**, *30*, 681–694. [[CrossRef](#)]
35. Boden, M.A. Creativity and Artificial Intelligence. *Artif. Intell.* **1998**, *103*, 347–356. [[CrossRef](#)]
36. Pasquale, F. *The Black Box Society: The Secret Algorithms That Control Money and Information*; Harvard University Press: Cambridge, MA, USA, 2015.
37. Hughes, T.P. *American Genesis: A Century of Invention and Technological Enthusiasm, 1870–1970*; University of Chicago Press: Chicago, IL, USA, 2004.
38. Backus, J.W. The FORTRAN Automatic Coding System. In *Proceedings of the Western Joint Computer Conference*; ACM: New York, NY, USA, 1957; pp. 188–198.
39. Simon, H.A. *The Sciences of the Artificial*, 3rd ed.; MIT Press: Cambridge, MA, USA, 1996.
40. Wiener, N. *Cybernetics: Or Control and Communication in the Animal and the Machine*; MIT Press: Cambridge, MA, USA, 1948.
41. Pask, G. The Architectural Relevance of Cybernetics. *Archit. Des.* **1969**, *39*, 494–496.
42. Banham, R. *The Architecture of the Well-Tempered Environment*; University of Chicago Press: Chicago, IL, USA, 1969.
43. Cross, N. *Design Methods: Strategies for Product Design*, 3rd ed.; Wiley: Chichester, UK, 2008.
44. Alexander, C. *Notes on the Synthesis of Form*; Harvard University Press: Cambridge, MA, USA, 1964.
45. Jones, J.C. *Design Methods*; Wiley: London, UK, 1970.
46. Broadbent, G. *Design in Architecture: Architecture and the Human Sciences*; Wiley: London, UK, 1973.
47. Mathews, S. *From Agit-Prop to Free Space: The Architecture of Cedric Price*; Black Dog Publishing: London, UK, 2007.
48. Chaillou, S. *ArchiGAN: Artificial Intelligence for Architectural Design*; Harvard Graduate School of Design: Cambridge, MA, USA, 2019.
49. Dillenburger, B.; Hansmeyer, M. Artificial Intelligence and Generative Design in Architecture. *Archit. Des.* **2023**, *93*, 82–89.
50. Duarte, J.P.; Beirão, J.N.; Montenegro, N.; Gil, J. Generative Housing Design Using Machine Learning. *Autom. Constr.* **2020**, *120*, 103–134.
51. Chaillou, S. *Artificial Intelligence and Architecture: From Research to Practice*; Birkhäuser: Basel, Switzerland, 2022.
52. Xu, X.; He, Z.; Feng, T.; Yuan, P.F. Generative AI in Architectural Design. In *AI for Architecture, Computational Design and Robotic Fabrication*; Springer: Singapore, 2024; pp. 3–15. [[CrossRef](#)]

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.