

Augmented Computational Design

Nourian, Pirouz; Azadi, Shervin; Uijtendaal, Roy; Bai, Nan

DOI

[10.1002/9781394172092.ch1](https://doi.org/10.1002/9781394172092.ch1)

Publication date

2024

Document Version

Final published version

Published in

Artificial Intelligence in Performance-Driven Design

Citation (APA)

Nourian, P., Azadi, S., Uijtendaal, R., & Bai, N. (2024). Augmented Computational Design. In N. Abbasabadi, & M. Ashayeri (Eds.), *Artificial Intelligence in Performance-Driven Design: Theories, Methods, and Tools* (pp. 1-30). Wiley. <https://doi.org/10.1002/9781394172092.ch1>

Important note

To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright

Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy

Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

Green Open Access added to TU Delft Institutional Repository

'You share, we take care!' - Taverne project

<https://www.openaccess.nl/en/you-share-we-take-care>

Otherwise as indicated in the copyright section: the publisher is the copyright holder of this work and the author uses the Dutch legislation to make this work public.

1

Augmented Computational Design

Pirouz Nourian¹, Shervin Azadi², Roy Uijtendaal³ and Nan Bai⁴

¹ University of Twente, Enschede, The Netherlands

² Eindhoven University of Technology, Eindhoven, The Netherlands

³ Nieman Raadgevende Ingenieurs, Utrecht, The Netherlands

⁴ Delft University of Technology, Delft, The Netherlands

Introduction

The core of the performance-driven computational design is to trace the sensitivity of variations of some performance indicators to the differences between design alternatives. Therefore, any argument about the utility of artificial intelligence (AI) for performance-based design must necessarily discuss the representation of such differences, as explicitly as possible. The existing data models and data representations in the field of architecture, engineering, and construction (AEC), such as computer aided design (CAD) and building information modelling (BIM), are primarily focused on geometrical representations of building elements and facilitating the process of construction management. Unfortunately, the field of AEC does not currently have a structured discourse based on an explicit representation of decision variables and desired outcomes. Specifically, the notion of design representation and the idea of data modeling for representing “what needs to be attained from buildings” is rather absent in the existing literature.

This treatise proposes a systematic view of the differences between design alternatives in terms of decision variables, be they spatial and nonspatial. Based on such an explicit formulation of decision variables, we set forth a framework for building and utilizing AI in (architectural) generative design processes for associating decision variables and outcomes of interest as performance indicators in a reciprocal relationship. This reciprocity is explained in terms of the duality between two quintessential problems to be addressed in generative design: the evaluation of design alternatives (mapping) and the derivation of design alternatives (navigation).

Starting with an explicit representation of a design space as an ordered pair of two vectors, one denoting decision variables and the other performance indicators, we put forth a mathematical framework for structuring data-driven approaches to generative design in the field of AEC. This framework highlights two major types of applications for AI in performance-driven design and their fusion: those capable of augmenting design evaluation procedures and those capable of augmenting design derivation procedures. Moreover, we introduce the

reciprocity between “flows” and “manifolds” as an intermediary notion for going beyond the so-called form-function dichotomy. Discussing these notions necessitates the introduction of a mathematical foundation for the framework rooted in multivariate calculus.

The main advantage of this explicit formulation is to enhance the explainability of AI when utilized in generative design by introducing meaningful and interpretable latent spaces based on the reciprocal relationship between manifolds and flows. The balance of predictive/deterministic power and interpretability/explainability is discussed in the concrete context of an illustrative example.

This chapter will introduce a chain of key concepts, starting with the notion of decision-making in design, the nature of design variables, the specifics of spatial decision variables, the notion of design space, and the two dual actions in the exploration of design spaces: mapping and navigating.

While the introduced framework is quite general, a particular class of Probabilistic Graphical Models (PGM), Bayesian Belief Networks (BBN), is introduced to provide a concrete illustrative example of the utility of AI in AEC. For a deeper insight into this particular approach to data-driven design, the readers are referred to two classical books on PGM: Pearl (1988) & Koller and Friedman (2009). The illustrative example is a BBN trained to make a data-driven replica of the building energy model used by the Dutch government in order to obtain a rough meta-model to be used in mass-scale policy analysis, such as advising the government on the relative utility of energy transition subsidies and planning measures. This example is chosen not because the BBNs are the most advanced models or the most accurate models for approximating such large functions. Instead, the choice is rather pragmatic in that this model has proven to be promising from the stance of predictive power while retaining a basic level of theoretical interpretability and intuitive appeal.

The chapter is structured as follows: we first present a historical context to establish the necessity of such a data-driven generative design framework; continue with conceptualizing and mathematically formulating the structure of the framework, dubbed as Augmented Computational Design (ACD); present an illustrative example demonstrating the utility of the framework; and conclude with a discussion on its outlook, open questions, and avenues for further research.

Background

Here, we revisit the utility of AI for data-driven generative design by highlighting some key gaps of knowledge in the field of AEC and briefly mentioning overarching frameworks in computational design and AI that can address these gaps.

Relevance of AI in AEC

The earliest attempts to enhance accountability and predictive power in computational design can be traced back to the notions of Scientific Architecture (Friedman 1975) and The Sciences of the Artificial (Simon 2019). Both of these seminal books explicitly discuss the necessity of forming some kind of a specific spatial and configurative form of design knowledge, the core of which boils down to being able to explicitly represent the main subject matter of spatial design as “spatial configurations.” One of the first phenomenological and

systemic descriptions of design processes explicitly referring to the notion of performance is the “Function, Behaviour, Structure” framework of Gero and Kannengiesser (2004), in which the overused notions of form and function are elaborated in terms of expected and required behavior/functionality from a system (dubbed as the function), its design as a form or configuration (dubbed as structure), and its performance (dubbed as behavior). The framework explicitly discusses the idea of design as a process of generating the representation of a spatial structure, and the difference between desired behavior and the actual behavior of the structure is discussed as the performance drive for the process. What can be observed in this phenomenological framework, predating most recent advancements in computational design, is the fundamental belief about the innate necessity of creativity in terms of the cognitive capability of designers for proposing structures capable of working as desired, based on some kind of tacit knowledge. Congruently, an anthropological description of design processes refers to the age-old duality between the form (structure) and function (purpose) of designed artifacts, and the fact that [in the absence of explicit knowledge and representation schemes], as Kroes (2010) has put it, designers are traditionally trained to produce solutions (draw them) through a “logical leap” often without even understanding or paying any attention to the design requirements or supposed levels of quality attainment. Suppose, we wanted to evaluate (compare) two different alternative designs for a hospital (Jia et al. 2023), or a home, the question is: How do we want to represent the designs digitally for a computer to evaluate them? Let us discuss an analogous example: if we wanted to compare two pieces of music in terms of their beauty, it would be very straightforward to digitize their notations and feed them to a machine, because the musical notation is already discretized (digitized), regardless if it is written on paper or etched on the cylinder of an old-fashioned winding music box (Zeng et al. 2021). However, doing the same, such as comparing two buildings, would be a much more difficult challenge, especially because there is currently no (discrete/textual) notation for spatial design that can capture the features of spatial configurations.

Instead of the extensive emphasis on the product of architecture as the shapes of buildings, we turn our attention to the processes of design and put a lens of “design as [discrete] decision-making” on the debate to avoid the common reduction of design to the production of design drawings. This view forms the basis of the generative design paradigm, as extensively articulated by Nourian, Azadi, and Oval (2023) and Veloso and Krishnamurti (2021). Similarly, the challenges, opportunities, and promising ways of utilization of AI, particularly deep-learning and generative models, for goal-oriented design explorations have been discussed extensively in Regenwetter, Nobari, and Ahmed (2022) and Regenwetter and Ahmed (2022).

Historical Context

In this section, we first give a very brief history of the most important and relevant developments in AI. Then, we lay the foundation for a formulation of architectural design as a matter of decision-making. We discuss the mathematical implications of this paradigmatic frame for generative design, elaborate on the notion of decision-making and the duality of derivation and evaluation problems, and discuss two statistical approaches to design: a possibilistic approach utilizing Fuzzy Logic or Markovian Design Machines and a probabilistic approach utilizing BBNs or Diffusion Models.

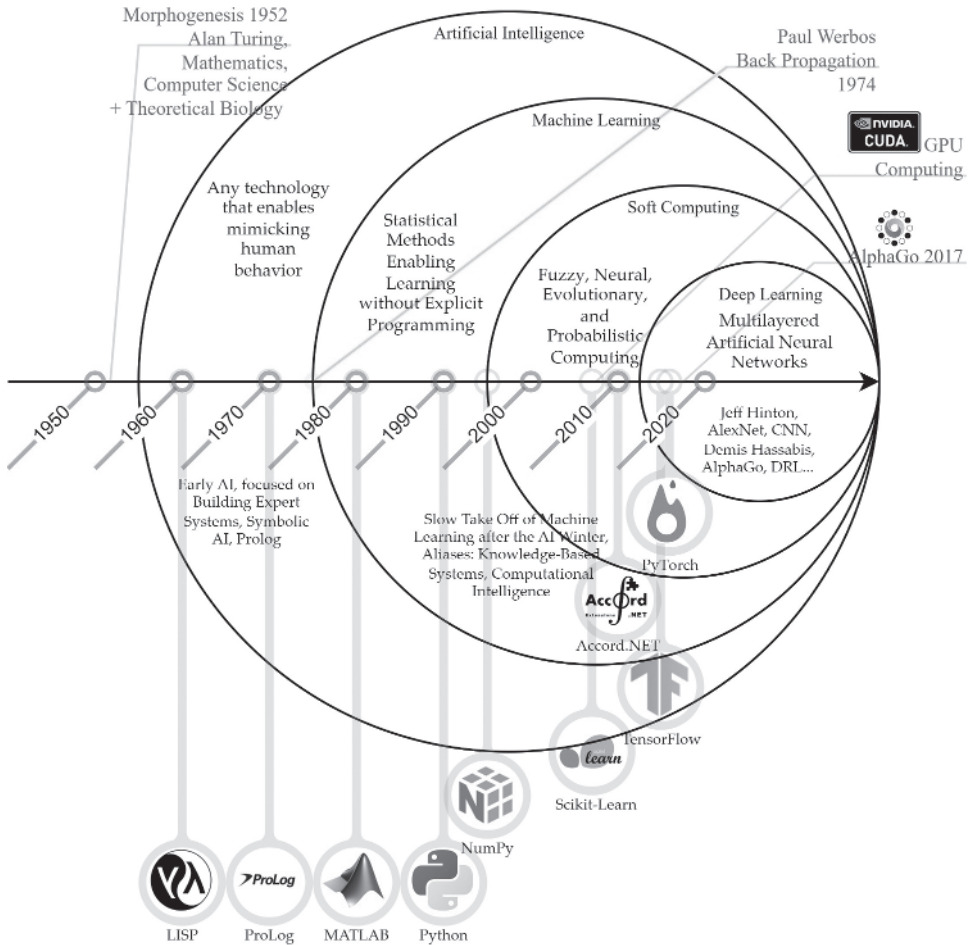


Figure 1.1 Highlights in the history of Artificial Intelligence.

We are currently witnessing an era of exponential success in the field of AI that has been evolving for more than 50 years (See Figure 1.1). Meanwhile, it is common knowledge that progress is slow in terms of innovation and scientific knowledge development in the field of AEC.

As extensively argued by Simon (1973) and Azadi and Nourian (2021), once an unambiguous language is adopted for discussing the classification of problems, we can see that many of the problems in AEC can be adequately (and possibly painstakingly) dealt with through conventional mathematics, physics, and computer science. In other words, the utility or the necessity of employing AI for dealing with problems that can be dealt with through conventional mathematical or computational procedures is not only pointless from a resource-efficiency stance but also questionable from the point of view of interpretability, transparency, and explainability. To assess the potential applications of AI in AEC regarding these questions, we highlight the history of AI (see Figure 1.1) and refocus on its scope (see Figure 1.2), at least as it could possibly pertain to AEC.

Once a problem is adequately formulated, two major determinants can be considered as to whether it would be sensible to apply AI or not: whether the data schemata of the problem

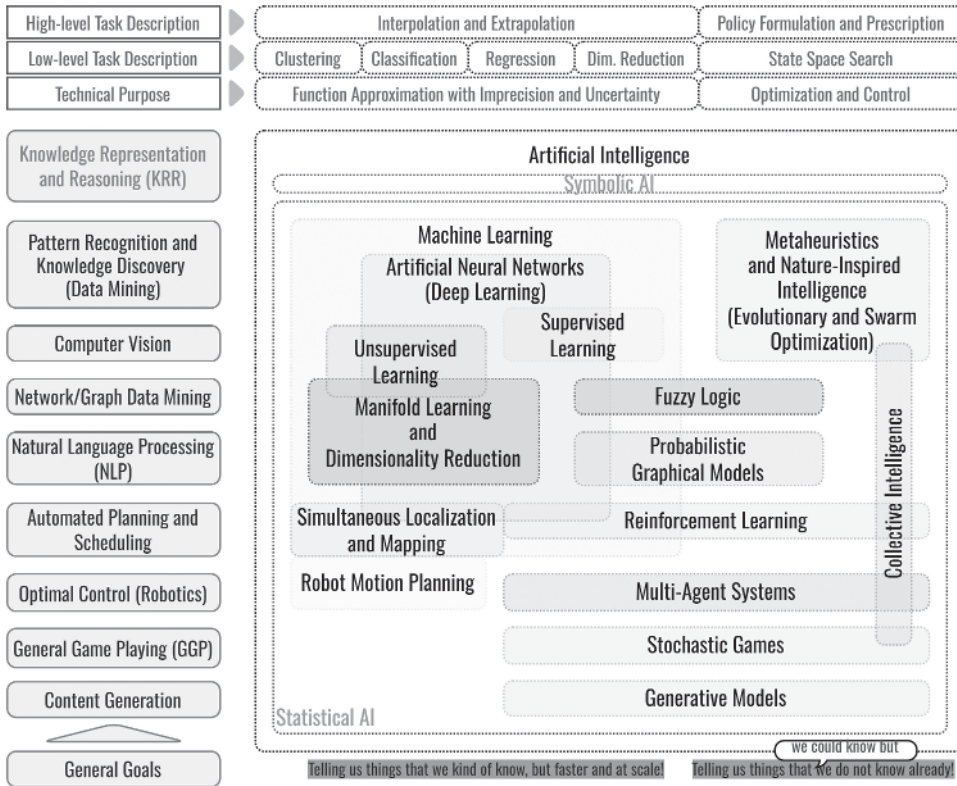


Figure 1.2 A Euler diagram of the scope of Artificial Intelligence.

are structured (vectorized) or unstructured (textual/visual), and whether the underlying associations between the inputs and outputs can be modeled through first principles (governing laws of physics, typically stated in differential equations), stochastic processes, or agent-based models. If the problem data are unstructured or the conventional modeling approaches do not have the capability of capturing the complex associations between the inputs and the outputs of interest, especially when interpretability can be sacrificed over the necessity of predictability, utilizing AI is quite sensible. The example that we discuss in this chapter may seem somewhat questionable according to these points; however, on the other hand, it is too overwhelmingly large and complex that no conventional approach can deal with it at the aimed level of abstraction. In this case, the ambition of the project is on such a high level of abstraction in policy analysis that the inaccuracies and ambiguities of the purely data-driven approach can be justified because of the insights that can be gained from the meta-statistical model.

Design as Decision-Making

The commonly overstated notions of difficulty or the ill-defined nature of design problems, as explained by Simon (1973), can be attributed to the fact that most design tasks are expected to produce a very concrete geometric description of an object to be built (the form), given only a very abstract description of what the object is supposed to be used for, how it should

work, and what would be desirable for it to achieve, all of which are often described quite vaguely (the function), by Kroes and Meijers (2006).

Hillier was one of the few shrewd theorists who understood that, at least after the separation of structural design from architecture in the 19th century (Giedion 2009), what distinguishes building buildings from architecture is the art and science of configuring spaces, as stated in “Space is the Machine,” (Hillier 2007). Once one realizes that the so-called architectural form is not only a single shape of an iconic object but also a set that includes the shapes of spaces and, eventually, the constituent segments of a building, we can distinguish the superior importance of spatial configurations. As obvious as it may sound, it seems to be necessary to emphasize that architectural design is not merely about sculpting a shape but configuring spaces to accommodate some human activities. This involves some puzzling tasks, such as packing, zoning, and routing spaces of various functions, which we hereinafter refer to as the task of configuring buildings (Azadi and Nourian 2021). For problems of shape and configuration to be transformed into decision problems, they need to be rigorously discretized. In short, we can call a massing problem a shape problem and a zoning problem a configuration problem.

The mainstay of the generative design paradigm is a rigorous reformulation of a design problem as a discrete topological decision problem rather than a geometrical problem (Nourian, Azadi, and Oval 2023). Therefore, discretization is the process of breaking down the integrated design problem into multiple smaller yet interdependent decision problems. An example of such discretization can be a voxel grid that provides a non-biased and homogeneous representation of spatial units, each of which poses a decision problem of function allocation (Nourian et al. 2016; Soman, Azadi, and Nourian 2022).

Moreover, to ensure the correspondence of these discrete decisions, we need to include the topological information about their neighborhood to represent their spatial interdependencies, similar to topology optimization (O’Shaughnessy, Masoero, and Gosling 2021). At the limit, such discretization can also be used to model a continuum of solutions and provide a frequency-based or spectral representation system, similar to the study by Marin et al. (2021), for spatial design, much like the musical notation that is based on notes.

Additionally, it is important to note that design decisions have a strong spatial dimension; however, they can include the social dimension to represent the preference of stakeholders and enable consensus-building (Bai et al. 2020). Given a view of design as a matter of decision-making, we can readily see two important types of practical questions that will shed light on the relevance of AI for decision-making:

1. How to map/learn the associations of hundreds or thousands of constituent choices of a compound design decision (function approximation and dimensionality reduction for ex-ante assessment of the impact of decisions)?
2. How to navigate a gigantic decision space with thousands of choices and their astronomically large combinations with a few important consequences in the picture?

The proposed notion of design as decision-making makes a point of departure for the rest of the chapter in that it highlights two equally essential problems of significance that can be tackled by AI and their duality: First, evaluation problems can be portrayed as mapping problems in Machine Learning (ML) and Deep Learning, where the approximation power of Artificial Neural Networks (ANN) can be exploited in regression and classification settings. Second, derivation problems can be portrayed as navigation problems in generative models, concerned with navigating from a low-dimensional representation of performance indicators toward disaggregated design decisions.

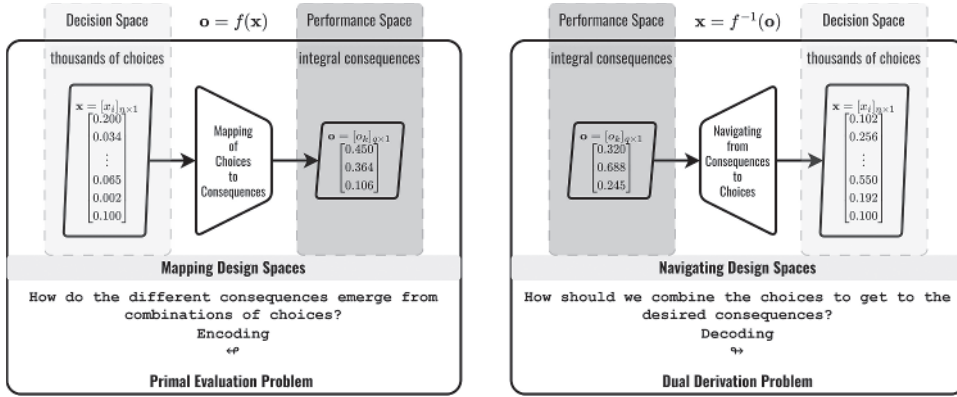


Figure 1.3 The duality of evaluation and derivation problems in generative design.

Encapsulating the complex and often nonlinear associations of many design decisions with a few outcomes of interest or performance indicators is here dubbed as a mapping problem. Inverting this map, as an approximated function (e.g. in the form of an ANN), can thus be viewed as an enhanced or augmented form of design, where the designer is navigated toward many small decisions just by pointing toward certain data points within a low-dimensional performance space (see Figure 1.3). It must be apparent that a navigation problem in this sense is much harder to solve, almost always impossible in the absolute sense, due to an arbitrarily large increase in information content and thus a combinatorial explosion of possibilities.

AI for Generative Design

Given the formulation of main generative design tasks as *mapping* and *navigating*, we focus on a particular set of AI methods that are distinguished for their relevance for these tasks in high-dimensional design decision spaces. More specifically, within the spectrum of generative design methods (Nourian, Azadi, and Oval 2023), we focus on data-driven *mapping* and *navigating* strategies. As shown in Figure 1.4, for brevity, we will only focus on the data-driven approaches to design on the right-hand side of the spectrum. Despite the other possible applications of (different kinds of) AI in this generative design spectrum, such as Reinforcement Learning in Policy-Driven design (playing design games), approximation of evaluation functions in topology or shape optimization, and Expert Systems in grammatical design, our framework here is focused on the statistical AI paradigm. So we only discuss the purely data-driven approaches to generative design.

Two subtle issues must be noted here: first, instead of discussing the utility of the wondrous application of generative models for the entertainment industry, we shall reflect on how the generative processes based on diffusion or dimensionality reduction can be controlled for attaining high-performance designs in an explainable manner. Second, model-driven approaches to performance-based generative design (topology optimization in particular) based on first principles, are already utilizing something important from the realm of nature-inspired computing called Hebbian Learning, which is already in the scope of (statistical) AI. This point, although important, generally interesting, and relatively unknown, falls outside the scope of this chapter.

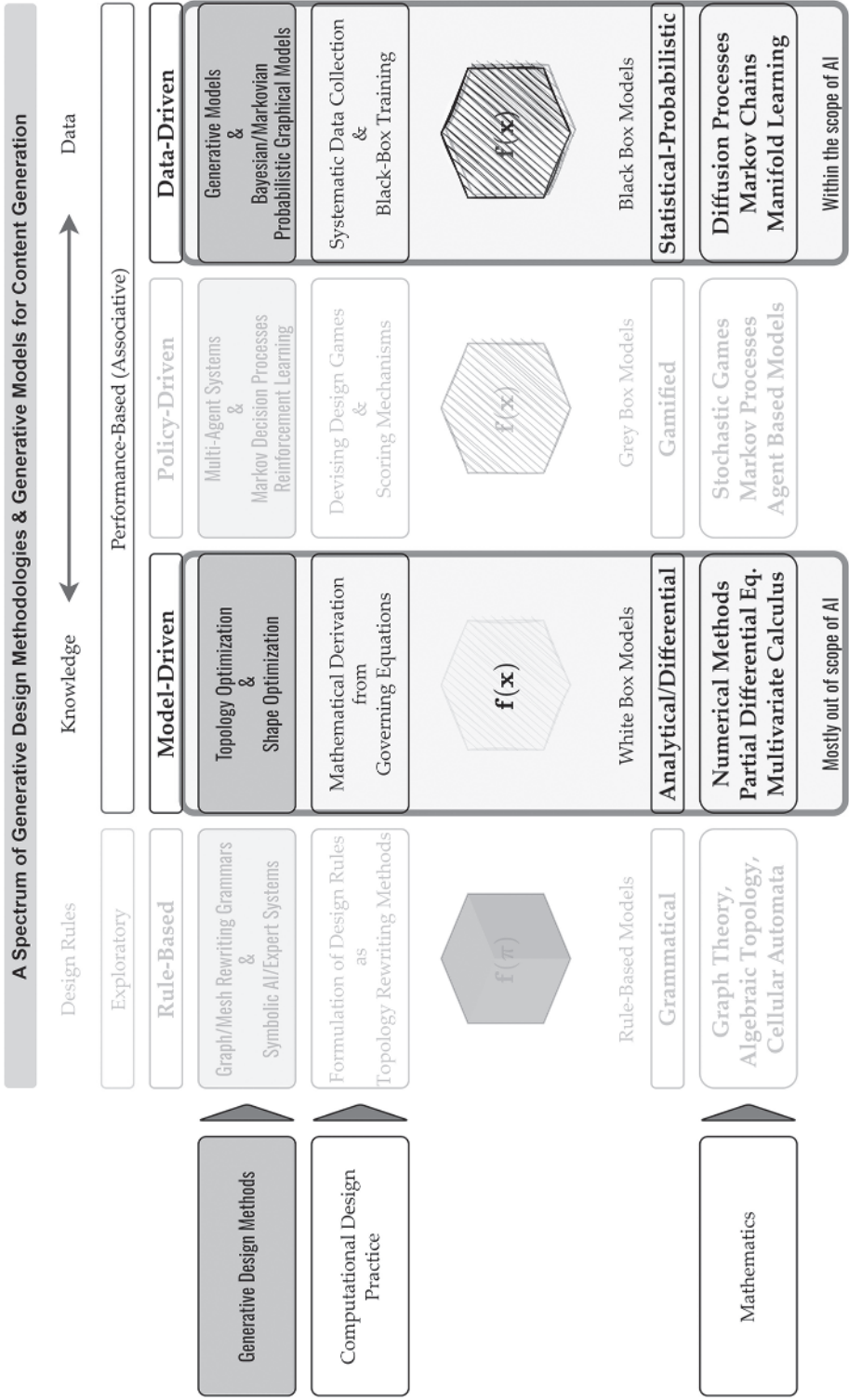


Figure 1.4 The spectrum of generative design methods and their relation to AI methods.

Framework

The emphasis on the decision-making approach to design entails that design tasks can be formulated as a set of (typically unstructured) questions about the form and materialization/construction of an object (a building) to be answered. In this chapter, we focus on the questions that pertain to form.

We propose a mathematical framework for generative design that relates multiple strands of work together. We use *design space* for referring to an ordered pair of two vector spaces: a decision space containing vectors or data points representing design configurations in the form of $\mathbf{x} \in (0, 1]^n$ and a performance/quality space containing vectors or data points representing combinations of outcomes of interest in the form of $\mathbf{o} \in [0, 1]^q$.

The mathematical lens that we shall put on the issue is to redefine both of these notions to provide a much more specific and workable idea for discussing the utility or futility of applying AI to design problem-solving. It is hopefully easy for the reader to accept that a regular discretization of so-called *design space* (which is an unfortunately common misnomer, but here somewhat pragmatically useful) provides a straightforward and simple discretization of design decisions as vectors in the form of $\mathbf{x} := [x_i]_{n \times 1} \in (0, 1]^n$ or $\mathbf{x} \in \{0, 1\}^n$, where n is the number of discrete cells in the design space, in which virtually any conceivable shape can be constructed at a certain level of resolution. Without loss of generality, the decision variables are not necessarily spatial and can be assumed to be relativized float variables within the range of minimum and maximum admissible parameter values of the functions that together result in the shape and configuration of a building. Even if a multicolor (multi-label, multi-functional) space is the subject of the design problem, then multiple categories/colors of such vectors can be seen together as a matrix of decision variables, in which rows of the matrix have to add up to 1 (see Figure 1.5).

Once this terminology is established, it is easy to observe that, in ML terms, the problem of performance-based design can be seen as two problems that are dual to one another: a multivariate regression problem for figuring out an approximation function that can map a few outputs to many inputs (referred here as mapping or the evaluation problem), and a pseudo-inversion problem for finding the combination of inputs that could result in desired output data points (referred to here as navigation or the derivation problem), as shown in Figure 1.3.

When approached as a data-driven problem-solving task, both problems are somewhat hard and impossible to solve in the absolute sense of the word, unless we think about them as loss minimization or approximation problems. The navigation/design problem is much harder than the mapping/evaluation problem. The main idea here is to advocate for training (fitting) meta-models (neural networks), to sets of sampled pairs of inputs and outputs, to first approximate a complex design space as a map between decision data points and performance data points, and then find the pseudo-inverse of this map or navigate it in the reverse direction to identify designs (decision data points) that perform in a desired way. In other words, mathematically, we look at the performance-based design process as a pairing between a decision space and a performance space, where a map is conceptualized as a function $f: (0, 1]^n \mapsto [0, 1]^q$ such that $\mathbf{o} = f(\mathbf{x})$. The pseudo-inverse map is thus dubbed as $f^{-1}: [0, 1]^q \mapsto (0, 1]^n$, such that $\mathbf{x} = f^{-1}(\mathbf{o})$.

For brevity and generalizability to nonspatial design problems, we will focus on massing problems, and exclude colored configuration problems out of the picture momentarily (see Figure 1.5 for the distinction).

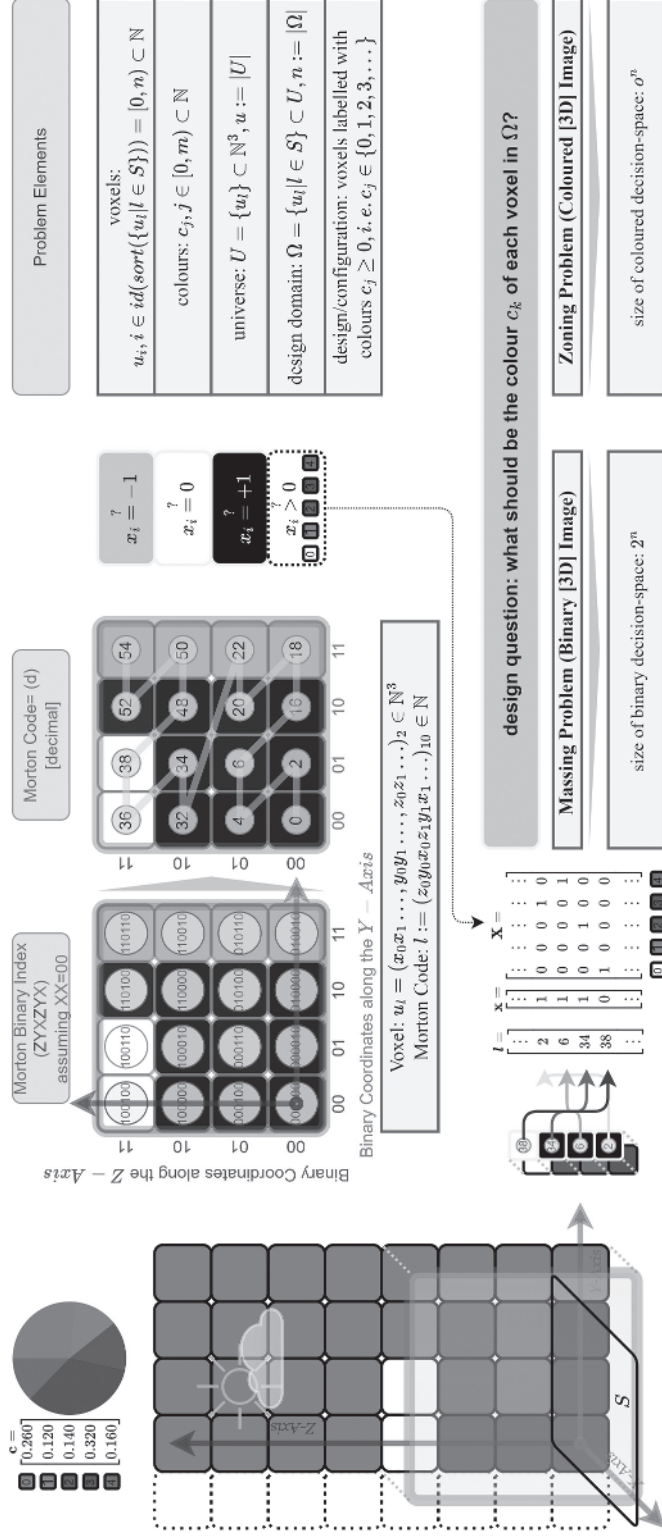


Figure 1.5 An illustrative discrete design domain and its associated decision space for shaping/massing and zoning/configuring problems.

Furthermore, by considering two abstract and high-level descriptions of a design task in our proposed regular discretization frameworks, we can formulate two mathematical tasks:

- Mapping Design Spaces: Approximating the function that models the associations between the many input design variables and a few outcomes of interest.
- Navigating Design Spaces: Approximating the inverse function that guides the generation of valid configurations in the decision space given desired data points in the performance space.

Design Space Exploration

Here, we explain the mathematical meaning of the two dual problems that, together, can be called design space exploration tasks: mapping and navigation.

Mapping

The problem of mapping associations between a large set of independent input decision variables and dependent output performance indicators is key to performance-driven design. Any explainable and accountable design methodology should have the capacity to guarantee the attainment of some quality or performance indicators. From a mathematical and statistical point of view, we might prefer to have an explainable and interpretable model of such relations that can be fitted into our data, or ideally, a simulation model to predict outputs from input data. However, in some cases, especially where a multitude of diverse quality/performance indicators are involved, and when one does not have an established basis for simulation modeling, statistical (data-driven) modeling seems to be the only option. And so, when the complexity of the model passes a certain threshold of nonlinearity and a multitude of inputs and outputs, we might prefer to trade interpretability for predictive power. That is exactly where ANNs as families of adjustable nonlinear functions stand out as viable function approximators. Training a network is practically a matter of minimizing a loss/error function by adjusting the parameters of a family of functions that are set out by the so-called architecture/structure of the ANN.

Even though this approximation is inherently nonlinear and global, it is illuminating to think of an alternative (locally) linear approximation based on the Jacobian Matrix. Suppose that $\mathbf{o} = f(\mathbf{x}) := [f_k(\mathbf{x})]_{q \times 1} = [f_k([x_i]_{n \times 1})]_{q \times 1}$ is a vector of multiple scalar functions of vector input variables. Then a basic idea of approximation is to approximate this function locally around an input data point by its Jacobian. This matrix operator gives the basis for a hyperplane equation that provides the n -dimensional Euclidean tangent space of the underlying function, similar to a multivariate regression hyperplane, although the latter would be fitted to the entire dataset.

Note that the ML task here would be a multivariate regression task in this case, that is, predicting the dependent given the independent variables. To understand the difficulty of

the mapping, consider that the Jacobian matrix $\mathbf{J} := [J_{k,i}]_{q \times n} = \left[\frac{\partial f_k}{\partial x_i} \right]_{q \times n} = [\nabla^T f_q]_{q \times 1}$ would

just provide the best local linear approximation of an otherwise globally nonlinear map from $\mathbb{R}^n$ to $\mathbb{R}^q$, that is, n decision variables to q quality criteria or performance indicators.

The Jacobian approximation is numerically computable provided the underlying function is smooth and differentiable. For brevity, as commonly done, we have omitted the fact that

the Jacobian can be evaluated at a certain input data point and that it is expected to be the best linear approximation of the function in question in the vicinity of that point. If we abbreviate the notation for the Jacobian as such a functional, then we can denote the approximate linear function at any given data point as follows, using the first-order Taylor Series expansion: $\mathbf{o}(\mathbf{x})|_{\mathbf{x} \sim \mathbf{x}_0} \approx \mathbf{J}(\mathbf{x}_0)(\mathbf{x} - \mathbf{x}_0)$, or simply put, as $\mathbf{o} \approx \mathbf{J}\mathbf{x}$, if we assume $\mathbf{x}$ to represent the vector of differences between the input data point with the center of the neighborhood.

The Jacobian approximation is also illuminating for another important reason: it allows us to approximate the Jacobian in a different sense, that is, in the sense of dimensionality/rank reduction using the Singular Value Decomposition (SVD) to see a clearer picture of the main factors playing the most significant roles in attaining the outcomes of interest; in other words, identifying the input variables to which the outcomes of interest are most sensitive. Even though we do not explicitly perform this operation in our demonstrative example using the SVD, it is still illuminating to see what SVD can do for this insightful approximation and dimensionality reduction for two reasons:

1. The SVD approximation of the Jacobian allows us to make a cognitive and interpretable map of the most important causes of the effects of interest.
2. The SVD approximation of the Jacobian allows us to conceptualize a pseudo-inverse function to navigate the design space from the side of performance data points.

The SVD (low-rank) approximation of the Jacobian matrix can be denoted as below:

$$\mathbf{J} := \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T$$

where, $\mathbf{U}_{q \times q} := [\mathbf{u}_k]_{1 \times q}$ and $\mathbf{V}_{n \times n} := [\mathbf{v}_i]_{1 \times n}$ are orthogonal matrices (i.e. $\mathbf{U}\mathbf{U}^T = \mathbf{I}_{q \times q}$ and $\mathbf{V}\mathbf{V}^T = \mathbf{I}_{n \times n}$), and $\mathbf{\Sigma}$ is a matrix of size $q \times n$ with only $p = \min\{q, n\}$ nonzero diagonal entries denoted as σ_c and called singular values, which are the square roots of the eigenvalues of both $\mathbf{J}^T\mathbf{J}$ and $\mathbf{J}\mathbf{J}^T$, sorted in descending order, see Martin and Porter 2012.

$$\mathbf{J} \approx \sum_{c \in [0, r)} \sigma_c \mathbf{u}_c \mathbf{v}_c^T$$

where $r \leq p$. It must be noted that the sum is not meant to be exhaustive; instead, the sum of the first significant terms achieves the purpose of dimensionality reduction of the decision space by showing a low-dimensional picture of the correlations between decision variables and their performance consequences. So, instead of decomposing the Jacobian up to p , we can choose to have a lower-dimensional approximation up to some arbitrary smaller number r .

Navigation

Navigating a high-dimensional design space from the side of the performance space toward the decision space for deriving design decisions (see Figure 1.3) is a very challenging task, almost always impossible in the absolute sense of solving the equation $\mathbf{J}\mathbf{x} = \mathbf{o}$, if the decision variables $\mathbf{x}$ are the unknowns.

It is easy to see that the Moore-Penrose pseudo-inverse of the approximated Jacobian matrix can be computed as a matrix of size $n \times q$ by easily using the SVD factorized matrix:

$$\mathbf{J}^\dagger := \mathbf{V} \mathbf{\Sigma}^\dagger \mathbf{U}^T$$

where, $\Sigma^\dagger$ is simply formed as a diagonal matrix of size $n \times q$ with the reciprocals of the singular values. Similarly, the approximate pseudo-inverse of the Jacobian can be computed as:

$$\mathbf{J}^\dagger \approx \sum_{c \in [0, r)} \sigma_c^{-1} \mathbf{v}_c \mathbf{u}_c^T$$

However, in the same way, a minimal-loss approximate solution exists for such equations when the matrix is rectilinear, $\mathbf{J}^\dagger \mathbf{o}$ is expected to be the least-square solution to the linearized Jacobian approximation of a navigation problem. Even though the system might theoretically have a solution, the odds of finding a unique solution are practically very skewed toward having an indeterminate system with many more inputs than outputs, and thus the system will have many approximate solutions rather than a unique exact solution.

This is of course in line with the intuition of most human beings about the inherent difficulty of design problems for which there is no unique solution. Note that in all these theoretical treatments, we implicitly assumed that all data points within the decision space correspond to valid designs, whereas, in reality, it might be more difficult to ensure finding valid solutions (feasible in the sense of complying with constraints), rather than good solutions. In other words, constraint solving tends to be more difficult than optimization within a feasible region of the decision space.

Spatial Design Variables

If the question of the design problem directly pertains to the shape of the configuration of an object, we can still construct decision variables to be handled within the proposed framework for mapping and navigating design spaces.

The idea of bringing spatial decision variables in a generative design process is to consider first the nature of the objects being designed as manifolds, that is, locally similar spaces (homeomorphic) to Euclidean spaces of low dimensions (2D planes or 3D hyperplanes), but globally more complex, possibly having holes, handles, and cavities (shells). Three types of these manifolds are of special interest for generative design, such as those that conduct walk flows (explicit or implicit pedestrian corridors in buildings and cities), light flows (rays of sunlight, sky-view, or other visibility targets), and force flows in structures. Our conceptual framework proposes that these flows are conducted within spatial manifolds, as below:

1. Walkable Space Manifolds (2D): Conduct walks (accessibility questions)
2. Air Space Manifolds (3D): Conduct light rays (visibility questions)
3. Material Space Manifolds (3D): Conduct forces (stability questions)

This consideration allows us to see that the way this object is supposed to function is largely determined by how this manifold is configured in that the way the manifold in question conducts the flows of walks in a walkable floor space, flows of light rays in a visible air space, or flows of forces in a reliable material space. Thus, we can highlight the specific concept of flow in a network representation, which is dual to the discrete representation of a manifold, as an unambiguous alternative intermediary instead of any vague notion of function to study and measure.

Apart from mathematical elegance, this approach also provides multiple computational advantages that are very much in line with the recent advancements in the field of generative models in AI. In a nutshell, the discrete representation of the so-called design space

provides a workable representation of not only all possible forms but also some inherent functional properties of the represented manifold that should logically determine how it could function as a building or a structure. The manifold representation can be mathematically denoted as a polygon mesh of vertices, edges, and faces $\mathcal{M} = (V, E, F)$ (for a two-manifold), or a polyhedral mesh of vertices, edges, faces, and cells $\mathcal{M} = (V, E, F, C)$ that can have a dual graph representation in the form of $\Gamma = (N, \Lambda)$.

This description should principally sound natural if we articulate the purpose of a design task as follows: finding the ideal form (configuration and shape) of a manifold to conduct some flows in a desirable pattern. In this way, we are diverting our attention from the containers of space (the building) into what it contains (i.e. the space and its spatial configuration). This change of focus allows us to see the direct correspondence between the so-called form and function of a design or, better put, the form (i.e. configuration and shape) and the expected quality/performance of a spatial configuration.

In what follows, we will go much beyond the vector data inputs consisting of only numerical variables, especially in the context of our illustrative example. In fact, without loss of generality, the ideas of mapping and navigating design spaces in an approximate sense go beyond decision variables pertaining to continuous decision variables and those pertaining to the spatial configuration and geometric shape of spatial manifolds. The same ideas can be applied to design problems that are about decision-making in a much more general sense, as discussed above. Note that the illustrative example that we have demonstrated at the end of the chapter has a heterogeneous mix of spatial and mostly nonspatial decision variables as well as a mix of categorical and numerical decision variables.

Statistical Approaches to Design

Among the statistical approaches to design, we can distinguish the possibilistic approaches from the probabilistic ones.

- Probabilistic approaches: BBN, Variational Auto-Encoders (VAE), and Diffusion Models
- Possibilistic approaches: Markovian Design Machines and Fuzzy Design (see MAGMA below)

Possibilistic Approach

The essence of the possibilistic approach to design using a multivalued or nonbinary logic framework for making design decisions, typically in the sense of making discrete choices about discrete segments of space; for example, the Markovian Design Machines of Batty (1974), the Spatial Agents Academy of Veloso and Krishnamurti (2020), and Multi-Attribute Gradient-Driven Mass Aggregation (MAGMA) through Fuzzy Logic, as introduced briefly in Nourian (2016) and Soman, Azadi, and Nourian (2021). Both of these methodologies apply nonbinary logic from a possibilistic point of view, in the sense that they take design inputs that are valued in the range of $[0, 1]$ but treat them as possibility measures rather than probability measures. The two big ideas behind these two methods are the utilization of Markov Chains, Markov Decision-Processes, and Fuzzy T-Norms for coping with uncertainty and human-like reasoning in simulated negotiations between spatial agents.

Probabilistic Approach

The probabilistic models briefly mentioned here are all related to the concept of conditional probability, the Bayes' theorem, and (generalized) stochastic processes that resemble Markov

Chains (Weng 2021; Nourian 2016). In a nutshell, the core of these models is about updating some posterior probabilities, indicating beliefs about the truth of some statements by prior probabilities multiplied by the likelihood of compelling evidence, scaled by the probability of the existence of the evidence. When probabilistic neuron-like nodes in PGM are combined, these new posterior probabilities or probability distributions can be fed into other layers of a network to create ANN architectures. A basic idea here is to gradually reduce the dimensionality of input data into an abstract low-dimensional representation (encoding, or mapping, albeit into a typically unintuitive and interpretable latent space), and then gradually use the inverse of the forward diffusion-like processes to denoise a vector in the low-dimensional hidden space. The latter process is called denoising or decoding, and it matches our description of navigation processes, albeit without direct control of the meaning of the latent space vectors. A breakthrough in this domain can come from enhancing the explainability of the latent space low-dimensional representations. This idea, however interesting, falls way outside the scope of this short treatise. Therefore, here we only provide a theoretical minimum for understanding the demonstrative example (i.e. a shallow BBN).

Demonstration

In this section, we will present a demonstration of the utility of the proposed framework to indicate how a discrete decision-making approach can facilitate generative design processes. As a disclaimer, it must be noted that this example is not chosen for technical reasons related to AI but rather due to its real, societal, and environmental importance for policy analysis concerning energy transition planning actions at the country level and sustainability strategies at the building level.

Case Study

Understanding the energy performance of architectural designs is crucial in ensuring a sustainable future. Building Energy Modeling (BEM) is a multipurpose approach used by designers and policymakers for checking building code compliance, certifying energy performance, subsidy policymaking, and building management. The Dutch government has recently introduced the NTA 8800 calculation model for quantitatively determining the energy performance and code compliance of buildings (“NTA 8800” 2022). The NTA 8800 aims to provide a transparent, verifiable, and enforceable building energy performance model, based on the European Energy Performance of Buildings Directive (EPBD), the European Committee of Standardization (CEN), and the Dutch Normalization Institute (NEN) published standards (“Nen 7120+C2:2012/A1:2017 NL” 2017). These regulations describe methods to calculate the energy performance of buildings, set energy requirements for new buildings, and make agreements about energy label obligations in existing buildings. The NTA 8800 only concerns building-related measures, as expressed in the EPBD, Annex A (Union 2021).

The NTA 8800 document has been implemented as an MS Excel tool by the Dutch government (commissioned by Nieman B.V. consultants). This calculation model translates the public European standard document into a calculation tool. The calculation tool is not publicly available, and it is not documented. Since we were given temporary and bounded access to this model, we chose to approximate it and construct a meta-model. The model consists of

269 unique input parameters about the spatial and technical building design configurations, based on which the model returns three scalar response values about the energy performance of the building design: BENG 1 (maximum permissible energy demand in kWh/m²y), BENG 2 (maximum permissible primary energy consumption in kWh/m²y), and BENG 3 (minimum permissible share of renewable energy use as a percentage). The acronym BENG refers to national performance indicators for Nearly Zero-Energy Buildings (Bijna Energie Neutrale Gebouwen in Dutch).

The NTA 8800 model has three main limitations: (1) it can only process and compute information about a single specific scenario at a time; (2) it returns scalar values about the energy performance that is untraceable to input parameters; and (3) missing input values could result in errors or nonrealistic response values. These three limitations make the model impractical for designers and policy analysts, particularly in the early stages of design. This impracticality is because, in conceptual design and policy analysis, designers need to (1) explore and iterate various options simultaneously; (2) need feedback on the degree of influence of each design decision; and (3) cannot provide detailed information yet about later-stage design choices, such as the technical systems.

The framework of ACD is particularly useful here as it allows us to relate the aggregated performance changes of the few NTA 8800 outputs of interest to the changes in the many design decision parameters of its input. In this particular case, we adopt a probabilistic meta-modeling (function approximation) approach based on the methodology suggested by Conti and Kaijima (2021).

Methodology

Meta-Modeling

Meta models are models that describe the structure, behaviors, or other characteristics of related models, providing a higher-level abstraction for constructing and interpreting complex numerical models that approximate more sophisticated models often based on simulations. A meta-model serves as a simplified, computationally efficient *model of the model* (Conti and Kaijima 2021), also referred to as a surrogate model (Kleijnen 1975). The process of creating a meta-model is referred to as meta-modeling (Gigch 1991). Some alternative meta-modeling techniques include interpolation methods, such as spline models (Barton 1998), polynomial regression (Kleijnen, n.d.), or Kriging (Ankenman, Nelson, and Staum 2010).

Within the ACD framework, such meta-models provide structured ways to perform the two most important tasks of the generative design: *mapping* and *navigating*.

In general, a standard meta-model can be described as: $\mathbf{o} = \mathbf{f}(\mathbf{x}) \simeq \mathbf{g}(\mathbf{x})$, where $\mathbf{o}$ is the aggregated simulation response, $\mathbf{f}$ denotes a computational simulation-based model conceptualized as a vector function, and $\mathbf{g}$ is the approximated model function (see Figure 1.3.) With this notation, the objective of meta-modeling is to build $\mathbf{g}$ in such a way that it produces reasonably close values of $\mathbf{o}$. In the case of ACD, meta-modeling can be adopted as a methodology of design *mapping* that provides a differentiable and ideally reversible $\mathbf{g}$ that can be used in the *navigating* process. In other words, the meta-modeling should structurally relate the choices and consequences in such a way that the choices can be derived from the desired consequences; hence, providing a data-driven basis for generative design. The next part demonstrates a probabilistic meta-modeling approach to navigation tasks in high-dimensional design decision spaces, based on the methodology introduced by Conti and Kaijima (2021).

Bayesian Belief Networks

A BBN is a kind of PGM that is effectively an ANN in the form of a Directed Acyclic Graph (DAG) with neuron-like nodes that can compute Joint Probability Distributions (JPDs) from input probability distributions or discrete Probability Density Functions (PDF), which is then attributed to an output probability distribution through a Conditional Probability Distribution (CPD) computing posterior probabilities/beliefs through the Bayes' theorem, hence the name Bayesian. The set of edges in a BBN forms the model architecture or structure that represents the particular probabilistic dependencies between discrete probability distributions attributed to the starting and ending nodes (Figure 1.6). This structure is typically set by the modeler based on their knowledge of the process, while the CPDs (transition probability matrices) are learned from experimental data. BBNs can help us semiautomatically reason about uncertain knowledge or data (Peng, Zhang, And Pan 2010). This makes it possible to perform probabilistic inference, such as computing the JPDs of some outputs (effects) given some inputs (causes). The name of these ANN comes from the idea of updating beliefs or hypotheses posterior to observing evidence; more precisely, utilizing Bayes' theorem for updating conditional probabilities in network structures, in a fashion similar to modeling and evaluating Markov Chains, with the difference that Markov Chains operate as uni-partite networks, each neuron in a BBN is a bipartite network coupled with an outer product calculator.

The neurons in a BBN consist of two layers. The first layer can be dubbed as a presynaptic layer that combines input discrete probability distributions (through an outer product) and forms a JPD and then flattens the JPD to form a vector-shaped probability distribution. The second, that is, the synaptic layer is a CPD, practically a rectangular probability transition matrix that maps this flattened JPD to the output probability distribution. A BBN then consists of such neurons connected in a DAG. Training a BBN means finding the entries of the CPD in such a way as to minimize the loss in the recovery of the output probability distribution from the input distributions. The appeal of BBNs is twofold: first, they allow the inclusion of expert knowledge and intuition into the network's architecture, and second, training of the network makes the network adapted to the objective data. In this case, we limit the architecture of the network to a single layer of neurons to keep the network invertible.

Workflow

Research by Conti and Kaijima (2018) illustrates the four process steps involved in developing a BBN meta-model. In this use case, we alter this methodology as creating a BBN with all

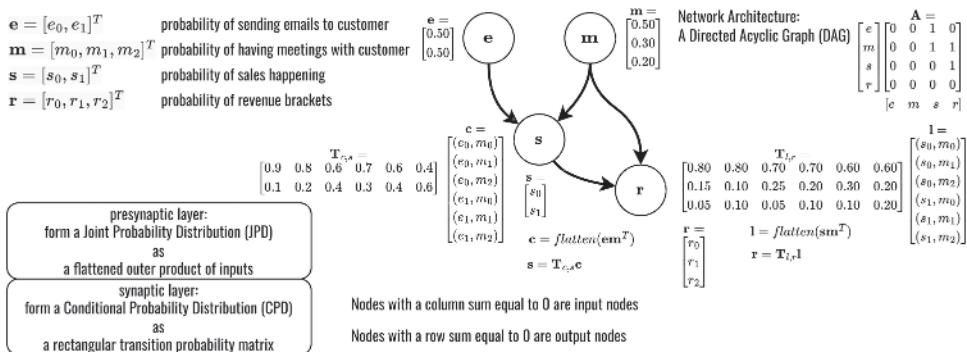


Figure 1.6 An illustrative example of a Bayesian Belief Network, eliciting the nature of nodes and the network architecture, an example inspired by Beaumont et al. (2021).

269 input parameters is infeasible. We add an intermediary step of sensitivity analysis to identify the most influential input parameters before constructing the BBN. Thus, we follow these steps in order (see Figure 1.7): (1) sample the input parameter space, (2) run simulations to generate the output values, (3) sensitivity analysis and selection of influential input parameters, (4) train the BBN, and finally, (5) evaluate the model's robustness. As highlighted by Conti and Kaijima (2021), it is important to model a shallow BBN as a complete bipartite graph connecting all input nodes to all output nodes, effectively limiting the topology to two layers. This would allow us to make a reversible approximation that can be used to derive the necessary input configuration for any desired performance output. Additionally, the fixed values can also include some of the input variables turning them into design constraints.

Step 1: Sampling the Parameter Space

We need to set up a Design of Experiment (DoE), to generate simulation data to study the relationships between various input variables and output variables (Hicks 1964). This experiment involves running several simulations at randomized input configurations (Sacks et al. 1989). Before running the simulation, it is important to carefully select a sampling method, to determine these input configurations, since the chosen strategy influences the quality of the meta-model (Fang, Li, and Sudjianto 2005). Since it is assumed that the decision space is unknown, the intention is to be as inclusive of all regions of the decision space as possible. The sampling algorithm should generate a well-varied response dataset that captures all the information about the relationships between the input parameters and responses. In this study, 20,000 quasi-random input samples were generated based on Sobol's sequences (Sobol' 1990) to ensure sample homogeneity.

Step 2: Run NTA 8800 Simulation Model

Vectorization is an important part of the ACD; we represent decision variables and outputs of interests as vectors (See Section Framework) Each sample point can be interpreted as a vector of scalar input values, $\mathbf{x}$. Each batch of such vectors is fed into the NTA 8800 model to generate the vector of corresponding building performance outputs, $\mathbf{o}$. After running the primary simulation model for the sampled input data points, the response data is collated and linked to the input samples to form an input-output dataset for regression modeling (as in ML).

Step 3: Sensitivity Analysis

The creation of a meta-model from 269 parameters, each with scalar input values, requires a simulation of all possible combinations (the number of options to the power of 269). Even limiting the number of options for each parameter to two, results in an immense number of possible combinations, calculated at 5.39×10^{80} . To contextualize the magnitude of this number, it is more than the estimated number of atoms in the observable universe.

The sheer magnitude of this number makes storage and training of BBNs infeasible. Therefore, in this study, we use global sensitivity analysis to apportion the uncertainty in outputs to the uncertainty in each input factor over their entire range. This allows us to remove the parameters with the lowest influence on energy performance. The sensitivity analysis method is implemented in the workflow based on the SALib library (Herman and Usher 2017).

This results in a meta-model with 15 parameters instead of 269, making it feasible to store and train the BBN; however, reducing the accuracy and scope of the model. However, the

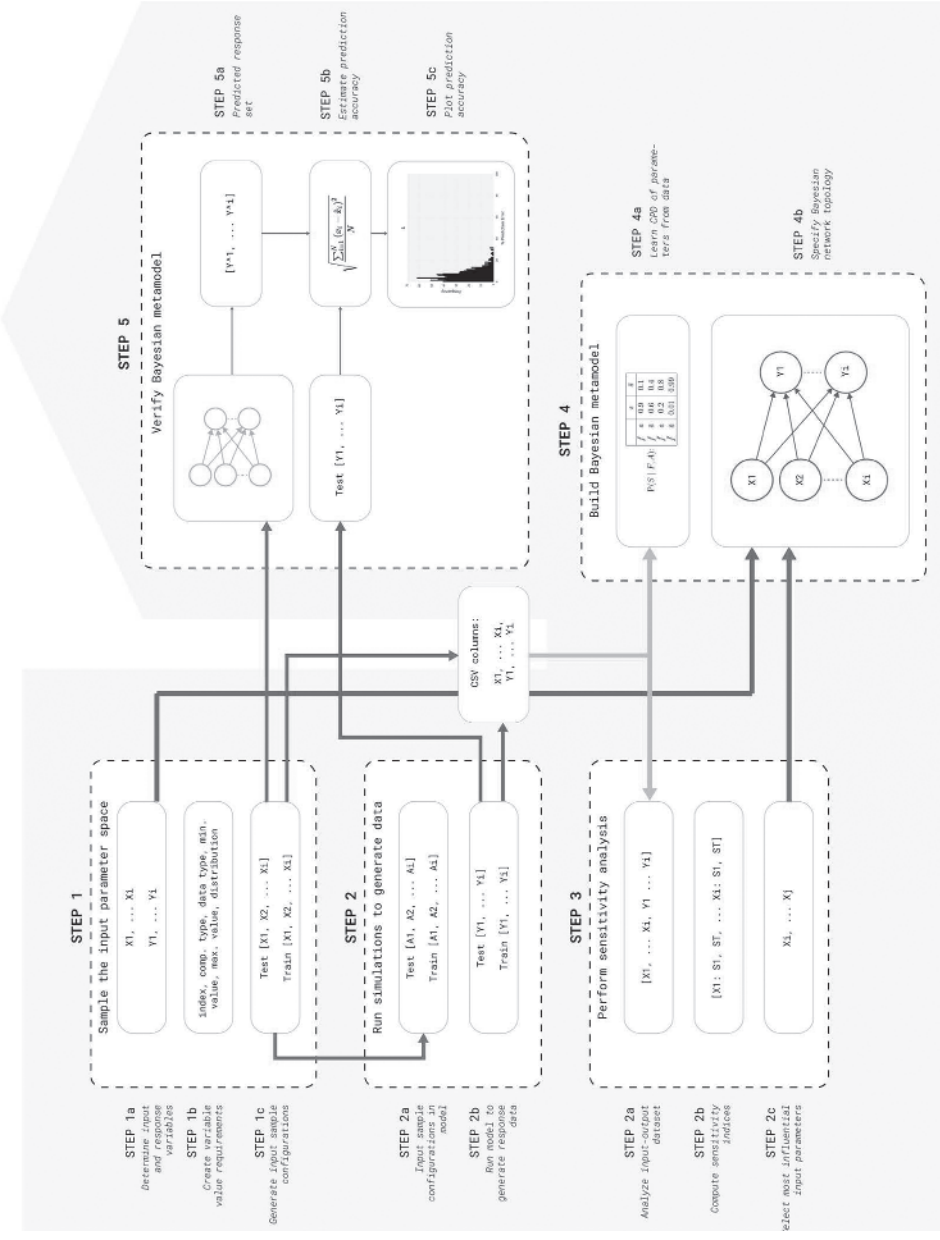


Figure 1.7 Overview of the workflow adopted from the framework. Source: Adapted from Conti and Kaijima (2018).

most influential 15 parameters are responsible for 90.45% to 92.30% of the final energy performance score. Hence, we decided on the inclusion of the 15 specific parameters to construct the BBN meta-model.

Step 4: Build a BBN Meta-model

Building a BBN meta-model is a process of associating the probabilistic relationships of inputs and outputs. These relationships may be characterized by a high degree of nonlinearity and possibly multiple interactions and correlations between model parameters. Consequently, there are two main steps in this process: (1) learning the network topology as a DAG structure and (2) estimating the CPD attributed to the neuron-like nodes of the network.

In this demonstration, we adhere to a particular network topology to ensure the reversibility of the trained model (Conti and Kaijima 2021). Accordingly, this BBN has only two layers: one corresponding to the input and one corresponding to the output. However, effectively, only a single layer of neurons operates in the middle of these two layers. In this case, the selected parameters from the sensitivity analysis results are represented by the input nodes, and the BENG 1, BENG 2, and BENG 3 parameters are the output nodes (see Figure 1.8). Therefore, we skip the topology learning step in the conventional BBN modeling because the topology of this particular network is assumed to be a complete bipartite DAG. In particular, we use the *pgmpy* Python package to model the network topology (Ankan and Panda 2015).

The next step is to estimate the CPDs for the nodes from the input–output dataset. The CPDs for the nodes can be directly learned from the input–output simulation data generated in steps 1 and 2, using the Maximum Likelihood algorithm. Additionally, we discretize each variable range into a fixed number of intervals. All numerical input distributions generated using a space-filling approach, such as Sobol’s sequence or Latin Hypercube, are sampled based on continuous ranges, and should therefore be discretized. Discretization is done by dividing the parameter interval over a fixed number of ranges between the minimum and maximum values.

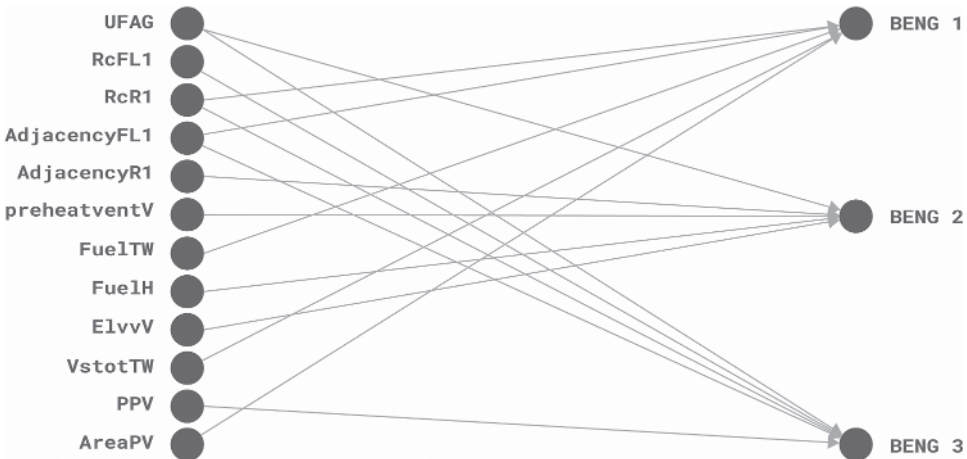


Figure 1.8 The single layer BBN: Right: BENG 1, BENG 2, and BENG 3 parameters are the output nodes; Left: Most sensitive input parameters as input nodes.

Step 5: Validating the Meta-Model

To assess how our trained BBN approximates the original NTA 8800 model, we use a cross-validation approach in combination with Normalized Root Mean Square Error (NRMSE) and Mean Absolute Percentage Error (MAPE) (James et al. 2013). The cross-validation splits the generated input–output dataset (step 2) into a *training set* and *testing set* before building the BBN (step 3). The BBN is trained on the *training set* and assessed based on the *testing set*. However, to obtain a more reliable estimate of the model’s performance, the dataset is split into several subsets or folds, with each fold used as both a training set and a testing set. This research adopts a k -fold cross-validation technique, where k refers to the number of groups that the data set is split into. We set $k = 10$ based on experimentation to ensure a low bias and a modest variance. The model is then trained on $k - 1$ of the folds, and the remaining fold is used for testing. This process is repeated k times, with each fold used for testing once. The model’s performance is then evaluated by averaging the performance across all k runs.

Following the approach suggested by Conti and Kaijima (2021), we computed the mean difference of the predicted and actual output values and normalized the RMSE values by dividing standard deviation to achieve NRMSE.

To calculate the accuracy of the meta-model, it is recommended to use multiple metrics to get a comprehensive evaluation of the model’s accuracy. Hence, NRMSE is combined with the MAPE metric. MAPE measures the average absolute percentage difference between predicted and actual values. It is a measure of the magnitude of the errors in the model’s predictions. Lower NRMSE and MAPE values indicate better model performance. The larger the error between the two, the higher the NRMSE and MAPE values will become. Therefore, the NRMSE and MAPE results will indicate how dispersed the prediction data is compared to the actual model response.

Results

This section presents the numerical results obtained from the experiment of NTA 8800 meta-model.

BBN Validation Results

Here, we elaborate on the results of the cross-validation technique in combination with NMSRE and MAPE based on the test dataset ($s = 1100$). The interpretation of what is considered an acceptable NRMSE and MAPE score depends on the specific problem and the context in which the meta-model is being used. In general, it is recommended to compare the NRMSE and MAPE scores of the meta-model with the baseline models and state-of-the-art models in the field. This can provide a benchmark for what is considered acceptable performance in the specific context of the problem.

In our case, the BBN does not compete with other models but rather competes with consulting building energy specialists in estimating building energy performance in the early design stages. However, to assess the proficiency of our model in capturing the underlying relationships using solely the 15 selected parameters, we employ the following benchmarks: The NRMSE values should be in the range of (0.20%, 0.60%) for the baseline, and in the range of (0.10%, 0.30%) for state-of-the-art (Bui et al. 2021) models. The MAPE values should be in the range of (0.10, 0.30) for the baseline, and in the range of (0.05, 0.15) for state-of-the-art models (Khan et al. 2021).

The NRMSE for the trained BBN are 0.82%, 1.52%, and 0.47% for BENG 1, BENG 2, and BENG 3, respectively. This indicates that, except for the BENG 3 indicator, the model's predictions are not accurate enough. The MAPE values for the trained BBN for BENG 1, BENG 2, and BENG 3 are 0.35, 0.28, and 0.33, respectively. This indicates that the predictions of the model are on the upper threshold of being acceptable as baseline models. The absolute prediction difference can be seen in Figure 1.9.

Toy Problem

Here, we present a test case that demonstrates the effectiveness of the BBN meta-model in building design. The study involves two toy problems that showcase the advantages and utilization of the meta-model. The toy problems address two common design challenges that cannot be solved using the currently available tools, such as the NTA 8800. The first problem involves predicting the BENG 1 energy performance of a typical Dutch dwelling during the early design stage. The spatial characteristics of the building are fed to the meta-model. As output, the meta-model returns a range and the confidence level of that range.

In this toy problem, the meta-model predicts the BENG 1 value to be within the range of $(0 - 50) \text{ kWh/m}^2 \cdot \text{y}$, with a 100% confidence level (see Figure 1.10). To validate this result, we cross-checked the predicted result with the final configuration of the dwelling using the original NTA 8800 model. The NTA 8800 model returns a value of $39.8 \text{ kWh/m}^2 \cdot \text{y}$, confirming the prediction capability of the meta-model.

The second problem reverses the first problem and involves the ex-ante determination of the most probable design configuration that satisfies a specific energy performance goal. In this example, the BENG 3 value of a typical Dutch dwelling design (35%) does not satisfy the minimal requirements (50%). Since this problem arises in the final design stage, some input parameters can no longer be changed. In this case, architects and engineers are limited to modifying only the area (AreaPV) and power (PPV) of the PV panels. Since the minimum required performance goal for BENG 3 is 50%, we set the goal value to a range of 60–80%. Given the binning approach employed, it should be noted that the AreaPV value of 5 depicted in the figure corresponds to a range of $[40, 50] \text{ m}^2$, while the PPV value of 5 corresponds to a PV Power range of $[200, 250] \text{ W/m}^2$. Accordingly, the meta-model advises increasing the PV area to $[40, 50] \text{ m}^2$, and the PV power to $[200, 250] \text{ W/m}^2$ (See Figure 1.10).

This discretization allows a clearer representation of the recommended parameter values within the specified ranges, facilitating the interpretation and practical implementation of the BBN meta-model outputs. Since these ranges are the maximum of both scales: the meta-models advise can be interpreted as maximizing the PV area and PV power to reach the goal BENG 3 value of 60% to 80%.

Discussion

In the end, to validate this result, we finish the loop by calculating the final configuration of the dwelling with the original NTA 8800 model. The NTA 8800 returns a value of 71%, confirming the reverse inference capability of the meta-model. These results, illustrated in a simple and digestible example, show how the BBN meta-model is capable of providing valuable insights and assisting architects and engineers in navigating the multidimensional decision space.

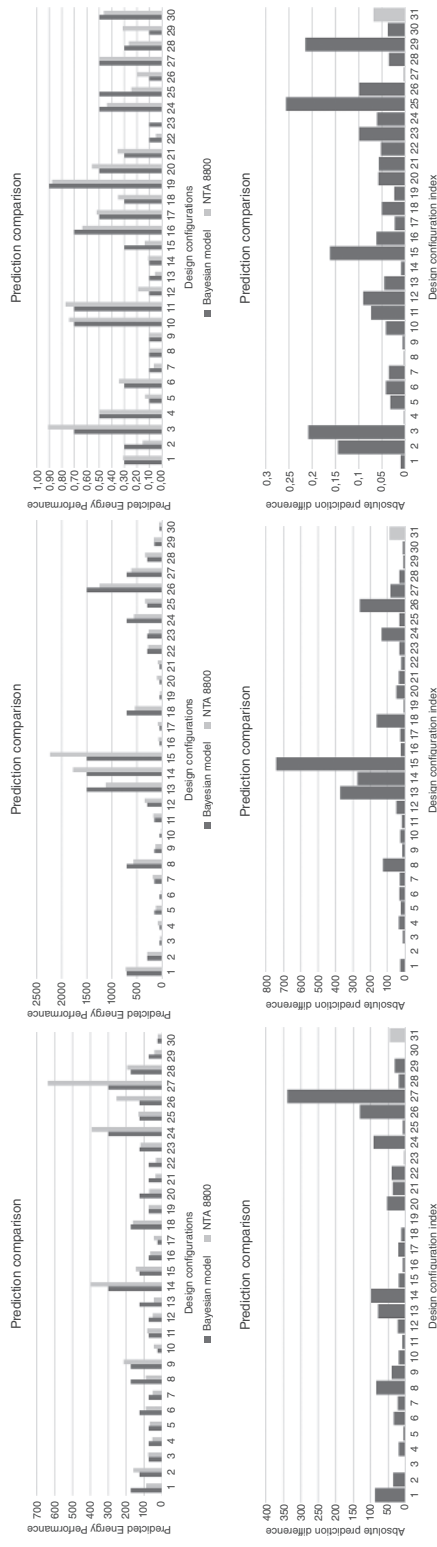


Figure 1.9 Histogram of BBN prediction and NTA 8800 outputs (top) and their comparison (bottom) for BENG1, BENG2, and BENG3, in order from left to right.

		AreaPV				
		1	2	3	4	5
PPV	1	0.033	0.038	0.039	0.041	0.044
	2	0.036	0.040	0.041	0.044	0.045
	3	0.044	0.043	0.041	0.039	0.045
	4	0.044	0.039	0.038	0.043	0.047
	5	0.047	0.047	0.047	0.047	0.048

Figure 1.10 Output Recommendations of the BBN Meta-model for Achieving BENG 3 Compliance.

By using the numerical DoE and Sensitivity Analysis, we are effectively conducting a dimensionality reduction task, similar to the low-rank SVD as introduced before. As mentioned earlier, the mention of this particular approach of ACD was to illustrate the utility of the framework with a concrete example in a societally relevant context where a ML approach to modeling can help make an otherwise very complicated simulation procedure to be approximately scaled up massively for policy analysis. Here, we discuss the potentials and shortcomings of the model and note the issues with this large-scale black-box approximation that require further investigation.

The existence of categorical variables in the inputs of the BBN limits the general applicability of ACD as it affects the smoothness and differentiability of the underlying function that is being approximated. However, for pragmatic reasons, we have ignored this issue to demonstrate the idea in a large-scale case.

Validation results

Compared to the NTA 8800, the Bayesian meta-model is capable of capturing the most important relationships between inputs and outputs. However, the difference between the meta-model's predictions and the NTA 8800 predictions can be rather high. This means that there is a large difference between the output of the meta-model and the NTA 8800. The NRMSEs of BENG 1 (0.82%) and BENG 2 (1.52%) show that the BBN is able to follow the NTA 8800 to some extent but is far from accurate, as NRMSE is greater than 0.5%. On the other hand, the model could predict BENG 3 (0.47) relatively accurately. This insufficient accuracy was expected as we have dictated a particular topology on the BBN while learning the network structure is an important step in constructing BBNs. This decision was made to enable the model to function in a bidirectional way: inference and reverse inference (i.e. evaluation and derivation in the terminology of our ACD framework).

Forward Inference

The trained BBN is now capable of inferring the outputs of interest given certain input configurations. This inference uses the learned CPDs to predict the most likely values for the

outputs. In this way, we can predict the energy performance of buildings, in a quick and intuitive way for ex-ante assessment based on a certain design configuration. In particular, this inference demonstrates the potential of a mapping described in Section 3.

Backward Inference

Since our BBN had only two layers in its network, it can be reversed. This means that, instead of presenting evidence to it, we can present the desired performance values and ask for the derivation of the particular configuration of inputs that will produce such an output. This can be done through the *Variable Elimination* module of the *pgmpy* (Ankan and Panda 2015). The same is also true for a combination of given inputs–outputs, meaning that evidence can be given for both inputs and outputs of the BBN. In such cases, the given inputs can also function as design constraints. The reverse inference demonstrates how we can utilize probabilistic models to navigate a decision space as explained in Section 3.

Augmenting

The Bayesian meta-model is capable of representing the input–output relationships in a bidirectional and probabilistic format, illustrating a complete example of mapping and navigating processes. However, the use of a subset of the most influential variables of the NTA 8800 limits the navigation to decision space made of the selected variables. Nevertheless, this selection was necessary to manage the computationally resource-intensive task of learning. Therefore, BBN does not compete with, or mimic the NTA 8800 model; rather, it complements it by increasing its accessibility and providing navigation capabilities. The result is a model that can augment the designers' intuition or experience and enhance the level of accuracy even in otherwise vague processes of policy formulation; for example, in assessing the potential efficacy of alternative subsidies and incentives for building renovation aimed at sustainable energy transition.

Outlook

The ACD framework and its constituent concepts can be best positioned within the context of performance-driven computational design and generative design. In particular, the idea of approximating complex and nonlinear functions for estimating measurable performance indicators from configurations of decision variables, even if referring to nonspatial decision variables, is generalizable to all areas of computer-aided design. However, such surrogate models are not to replace simulation models based on first principles, as they cannot match their transparency and explainability. Nevertheless, in cases where one needs to estimate the effects of design decisions on human factors, ergonomics, or combinations of many different types of governing equations, an estimation model trained from actual data can be of utility in that it provides a basis for comparisons in the absence of analytical knowledge. In other words, the utility of ANNs for *mapping* associations between decision data points and performance data points is apparent.

The *navigation* problem, on the other hand, is much harder, philosophically, technically, and mathematically for being solved in any sense. The real advantage of an AI framework in dealing with a design space navigation problem can be attained if the latent space of the model reveals interpretable information or if it is at least coupled with a sensible low-dimensional space. If the latent space of, for example, an Auto Encoder (Marin et al. 2021) is

understandable as a low-dimensional vector space (as an endpoint of the *mapping* and the start point of the *navigating* processes), it can be used not only to guide the navigation process but also to gain insight into which design variables are more important in determining the attainment levels of outcomes of interest. In other words, even though it appears that in the mapping process, the information content of the decision data points is gradually reducing, one can think of this process as a distillation of an elixir from a large data point that makes the information richer from a human perspective.

In light of this, the major advantages of the proposed framework are twofold: First, it provides an elegant framework for applying AI in computational design in the presence of many complex quality criteria; and second, it provides an elegant framework for designing spatial manifolds very much like the methodology of electrical engineering in designing electronic circuits and systems for signal processing. The latter point requires much more space for discussing the theoretical minimum for such an approach to design from a signal processing standpoint. In short, however, we can briefly mention that the idea of defining a central representation of a configuration as a discrete manifold provides for directly modeling the functionality of the spatial manifold with respect to the flows of walks, light rays, or forces not only from the point of view of spatial movement but also much more elegantly and efficiently in the frequency or spectral domain (which can be attained using Discrete Fourier Transform or Spectral Mesh Analysis). One fundamental idea of analog circuit design from a signal processing point of view is that of designing passive “filters,” whose properties can much better be understood in the so-called frequency domain analyses put forward by Fourier and Laplace transforms of the so-called transfer functions of the Resistor, Self-Induction Loop, Capacitor (RLC) circuits. This approach to circuit design can be traced back to the ideas and propositions of Oliver Heaviside (1850–1925), a self-educated pioneer of electrical engineering. Arguably, this frequency-based outlook, relating to the spectrum of eigen frequencies of vibration of shapes (also identifiable as a spectral approach), has revolutionized the formation of the field of electronics and thus contributed significantly to the development of AI as we know it today. Identifying spectral latent spaces and associating them with low-dimensional performance spaces and latent spaces of ANNs is a topic that calls for further theoretical research and computational experimentation.

Acronyms

Acronym	Term
ACD	Augmented Computational Design
AEC	Architecture, Engineering, and Construction
AI	Artificial Intelligence
ANN	Artificial Neural Networks
BBN	Bayesian Belief Networks
BEM	Building Energy Modeling
BENG	Bijna Energie Neutrale Gebouwen: Nearly Zero-Energy Buildings
BIM	Building Information Model
CAD	Computer-Aided Design

Acronym	Term
CEN	Comité Européen de Normalisation: European Committee of Normalization
CPD	Conditional Probability Distribution
DAG	Directed Acyclic Graph
DoE	Design of Experiment
EPBD	European Energy Performance of Buildings Directive
JPD	Joint Probability Distributions
MAPE	Mean Absolute Percentage Error
MAGMA	Multi-Attribute Gradient-Driven Mass Aggregation
ML	Machine Learning
NEN	Nederlandse Norm: Royal Dutch Standardization Institute
NRMSE	Normalized Root Mean Square Error
NTA 8800	Nederlandse Technische Afspraak (Dutch Technical Agreement)
PDF	Probability Density Functions
PGM	Probabilistic Graphical Models
SVD	Singular Value Decomposition
VAE	Variational Auto-Encoders

Notations

Notation	Name	Definition
$\mathbf{x}$	Design/decision space	$\mathbf{x} \in (0, 1]^n$; each x_i corresponds to a single spatial decision variable
$\mathbf{o}$	Performance space	$\mathbf{o} \in (0, 1]^q$; each o_k corresponds to an objective or outcome of interest
$\mathbf{o} = f(\mathbf{x}) := [f_k(\mathbf{x})]_{q \times 1}$	Map from design to performance	$f: (0, 1]^n \mapsto [0, 1]^q$; representing a meta-model that approximately maps the decision space to the performance space
$\mathbf{x} = f^{-1}(\mathbf{o})$	Map from performance to design	$f^{-1}: [0, 1]^q \mapsto (0, 1]^n$; pseudo-inverse of a meta-model that approximately maps the performance space to the decision space
$\mathbf{J} := [J_{k,i}]_{q \times n}$	Jacobian matrix of f	$[J_{k,i}]_{q \times n} = \left[\frac{\partial f_k}{\partial x_i} \right]_{q \times n} = \left[\nabla^T f_q \right]_{q \times 1}$
$\mathbf{U}_{q \times q} := [\mathbf{u}_k]_{1 \times q}$	Matrix of left singular vectors	$\mathbf{U}\mathbf{U}^T = \mathbf{U}^T\mathbf{U} = \mathbf{I}_{q \times q}$; ordered by importance
$\mathbf{V}_{n \times n} := [\mathbf{v}_i]_{1 \times n}$	Matrix of right singular vectors	$\mathbf{V}\mathbf{V}^T = \mathbf{V}^T\mathbf{V} = \mathbf{I}_{n \times n}$; ordered by importance
$\mathbf{\Sigma}_{q \times n} := [\mathbf{u}_k]_{1 \times q}$	Matrix of singular values	$\mathbf{\Sigma}$ is an $q \times n$ rectangular diagonal matrix with non-negative real numbers on the diagonal ordered by importance, i.e. singular values σ_c , $c \in [0, \min\{q, n\})$

References

- Ankan, A. and Panda, A. (2015). Pgmpy: Probabilistic Graphical Models Using Python. In: *Proceedings of the 14th Python in Science Conference (Scipy 2015)*, vol. 10. Citeseer.
- Ankenman, B., Nelson, B.L., and Staum, J. (2010). Stochastic Kriging for Simulation Metamodeling. *Operations Research* 58 (2): 371–382. <https://doi.org/10.1287/opre.1090.0754>.
- Azadi, S. and Nourian, P. (2021). GoDesign: A Modular Generative Design Framework for Mass-Customization and Optimization in Architectural Design. In: *Towards a New, Configurable Architecture*, vol. 1, 285–294. Novi Sad, Serbia: CUMINCAD.
- Bai, N., Azadi, S., Nourian, P., and Roders, A.P. (2020). Decision-Making as a Social Choice Game. In: *Proceedings of the 38th eCAADe Conference*, vol. 2, 10.
- Barton, R.R. (1998). Simulation Metamodels. In: *1998 Winter Simulation Conference. Proceedings (Cat. No.98CH36274)*. IEEE <https://doi.org/10.1109/wsc.1998.744912>.
- Batty, M. (1974). A Theory of Markovian Design Machines. *Environment and Planning B: Planning and Design* <https://doi.org/10.1068/b010125>.
- Beaumont, P., Horsburgh, B., Pilgerstorfer, P. et al. (2021). CausalNex [Computer software]. <https://github.com/quantumblacklabs/causalnex>.
- Bui, V., Le, N.T., Nguyen, V.H. et al. (2021). Multi-Behavior with Bottleneck Features LSTM for Load Forecasting in Building Energy Management System. *Electronics* 10 (9): 1026. <https://doi.org/10.3390/electronics10091026>.
- Conti, Z.X. and Kaijima, S. (2018). A Flexible Simulation Metamodel for Exploring Multiple Design Spaces. In: *International Association for Shell; Spatial Structures (IASS)*. <https://doi.org/10.13140/RG.2.2.23313.53600>.
- Conti, Z.X. and Kaijima, S. (2021). Explainable ML: Augmenting the Interpretability of Numerical Simulation Using Bayesian Networks. In: *The Routledge Companion to Artificial Intelligence in Architecture*, 315–335. Routledge.
- Fang, K.-T., Li, R., and Sudjianto, A. (2005). *Design and Modeling for Computer Experiments*. Chapman: Hall/CRC <https://doi.org/10.1201/9781420034899>.
- Friedman, Y. (1975). *Toward a Scientific Architecture*. First American Edition. Cambridge, Mass: MIT Press.
- Gero, J.S. and Kannengiesser, U. (2004). The Situated Function–Behaviour–Structure Framework. *Design Studies* 25 (4): 373–391.
- Giedion, S. (2009). *Space, Time and Architecture: The Growth of a New Tradition*. Harvard University Press.
- van Gigch, J.P. (1991). *System Design Modeling and Metamodeling*. Language of Science: Plenum <https://books.google.nl/books?id=M5mD0ZZcwaEC>.
- Herman, J. and Usher, W. (2017). SALib: An Open-Source Python Library for Sensitivity Analysis. *The Journal of Open Source Software* 2 (9): 97. <https://doi.org/10.21105/joss.00097>.
- Hicks, C.R. (1964). *Fundamental Concepts in the Design of Experiments*. New York: Holt, Rinehart; Winston.
- Hillier, B. (2007). *Space Is the Machine: A Configurational Theory of Architecture*. Space Syntax.
- James, G., Witten, D., Hastie, T., and Tibshirani, R. (2013). *An Introduction to Statistical Learning: With Applications in R*, Springer Texts in Statistics. Springer New York https://books.google.nl/books?id=qcI_AAAAQBAJ.
- Jia, Z., Nourian, P., Luscure, P., and Wagenaar, C. (2023). Spatial Decision Support Systems for Hospital Layout Design: A Review. *Journal of Building Engineering* 106042.

- Khan, A.N., Iqbal, N., Ahmad, R., and Kim, D.-H. (2021). Ensemble Prediction Approach Based on Learning to Statistical Model for Efficient Building Energy Consumption Management. *Symmetry* 13 (3): 405. <https://doi.org/10.3390/sym13030405>.
- Kleijnen, J.P.C. (1975). A Comment on Blannings Metamodel for Sensitivity Analysis: The Regression Metamodel in Simulation. *Interfaces* 5 (3): 21–23. <https://doi.org/10.1287/inte.5.3.21>.
- Kleijnen, J.P.C. (n.d.). Low-Order Polynomial Regression Metamodels and Their Designs: Basics. In: *International Series in Operations Research and Management Science*, 15–71. Springer US https://doi.org/10.1007/978-0-387-71813-2_2.
- Koller, D. and Friedman, N. (2009). *Probabilistic Graphical Models: Principles and Techniques*. In: *Adaptive Computation and Machine Learning*. Cambridge, MA: MIT Press.
- Kroes, P. (2010). Engineering and the Dual Nature of Technical Artefacts. *Cambridge Journal of Economics* 34 (1): 51–62.
- Kroes, P. and Meijers, A. (2006). The Dual Nature of Technical Artefacts. *Studies in History and Philosophy of Science* 37 (1): 1–4.
- Marin, R., Rampini, A., Castellani, U. et al. (2021). Spectral Shape Recovery and Analysis Via Data-Driven Connections. *International Journal of Computer Vision* 129 (10): 2745–2760. <https://doi.org/10.1007/s11263-021-01492-6>.
- Martin, C.D. and Porter, M.A. (2012). The Extraordinary SVD. *The American Mathematical Monthly* 119 (10): 838–851.
- Nen 7120+C2:2012/A1:2017 NL. 2017. *NEN*. <https://www.nen.nl/nen-7120-c2-2012-a1-2017-nl-229670>.
- Nourian, P. (2016). *Configraphics: Graph Theoretical Methods for Design and Analysis of Spatial Configurations*. *TU Delft Open* <https://doi.org/10.7480/isbn.9789461867209>.
- Nourian, P., Azadi, S., and Oval, R. (2023). Generative Design in Architecture: From Mathematical Optimization to Grammatical Customization. In: *Computational Design and Digital Manufacturing* (ed. P. Kyratsis, A. Manavis, and J. Paulo Davim), 1–43. Cham: Springer International Publishing https://doi.org/10.1007/978-3-031-21167-6_1.
- Nourian, P., Gonçalves, R., Zlatanova, S. et al. (2016). Voxelization Algorithms for Geospatial Applications: Computational Methods for Voxelating Spatial Datasets of 3D City Models Containing 3D Surface, Curve and Point Data Models. *MethodsX* 3 (January): 69–86. <https://doi.org/10.1016/j.mex.2016.01.001>.
- “NTA 8800.” 2022. *Gebouwenenergieprestatie (EPG)*. <https://www.gebouwenenergieprestatie.nl/bepalingsmethode>.
- O’Shaughnessy, Connor, Enrico Masoero, and Peter D. Gosling. 2021. “Topology Optimization Using the Discrete Element Method. Part 1: Methodology, Validation, and Geometric Nonlinearity.” Preprint. *engrXiv*. <https://doi.org/10.31224/osf.io/c6ymn>.
- Pearl, J. (1988). *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan kaufmann.
- Peng, Y., Zhang, S., and Pan, R. (2010). Bayesian Network Reasoning With Uncertain Evidences. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems* 18 (05): 539–564. <https://doi.org/10.1142/s0218488510006696>.
- Regenwetter, L. and Ahmed, F. (2022). Towards Goal, Feasibility, and Diversity-Oriented Deep Generative Models in Design. *arXiv* <http://arxiv.org/abs/2206.07170>.
- Regenwetter, L., Nobari, A.H., and Ahmed, F. (2022). Deep Generative Models in Engineering Design: A Review. *Journal of Mechanical Design* 144 (7): 071704. <https://doi.org/10.1115/1.4053859>.

- Sacks, J., Welch, W.J., Mitchell, T.J., and Wynn, H.P. (1989). Design and Analysis of Computer Experiments. *Statistical Science* 4 (4): <https://doi.org/10.1214/ss/1177012413>.
- Simon, H.A. (1973). The Structure of Ill Structured Problems. *Artificial Intelligence* 21.
- Simon, H.A. (2019). *The Sciences of the Artificial, Reissue of the Third Edition with a New Introduction by John Laird*. MIT press.
- Sobol', I'y.M. (1990). On Sensitivity Estimation for Nonlinear Mathematical Models. *Matematicheskoe Modelirovanie* 2 (1): 112–118.
- Soman, A., Azadi, S., and Nourian, P. (2022). DeciGenArch: A Generative Design Methodology for Architectural Configuration via Multi-Criteria Decision Analysis. In: *Proceedings of eCAADe 2022*, forthcoming. Education; research in Computer Aided Architectural Design in Europe. <https://www.overleaf.com/project/60f1ae165edde605561c28e1>.
- Union, European (2021). Proposal for a Directive of the European Parliament and of the Council on the Energy Performance of Buildings (Recast). *Lex - 52021PC0802 - En - EUR-Lex* <https://eur-lex.europa.eu/legal-content/EN/ALL/?uri=CELEX%3A52021PC0802>.
- Veloso, Pedro, and Ramesh Krishnamurti. 2020. “An Academy of Spatial Agents: Generating Spatial Configurations with Deep Reinforcement Learning.” In *eCAADe: Anthropologic - Architecture and Fabrication in the Cognitive Age*.
- Veloso, P. and Krishnamurti, R. (2021). Mapping Generative Models for Architectural Design. In: *The Routledge Companion to Artificial Intelligence in Architecture*, 29–58. Routledge.
- Weng, L. (2021). What Are Diffusion Models? *Lilianweng.github.io*, July. <https://lilianweng.github.io/posts/2021-07-11-diffusion-models/>.
- Zeng, M., Xu, T., Wang, R. et al. (2021). MusicBERT: Symbolic Music Understanding with Large-Scale Pre-Training. In *ACL-IJCNLP 2021*: <https://www.microsoft.com/en-us/research/publication/musicbert-symbolic-music-understanding-with-large-scale-pre-training/>.