Speculative hybrids

Investigating the generation of conceptual architectural forms through the use of 3D generative adversarial networks

Pouliou, Panagiota; Horvath, Anca Simona; Palamas, George

Published in:
International Journal of Architectural Computing

DOI (link to publication from Publisher):
10.1177/14780771231168229

Publication date:
2023

Citation for published version (APA):
Pouliou, P., Horvath, A. S., & Palamas, G. (2023). Speculative hybrids: Investigating the generation of conceptual architectural forms through the use of 3D generative adversarial networks. *International Journal of Architectural Computing*, *21*(2), 315-336. https://doi.org/10.1177/14780771231168229

Check for updates

Special Issue

IJAC

International Journal of
Architectural Computing
2023, Vol. 0(0) 1–22

DOI: 10.1177/14780771231168229
journals.sagepub.com/home/jac
SAGE

Speculative hybrids: Investigating the generation of conceptual architectural forms through the use of 3D generative adversarial networks

Panagiota Pouliou[1], Anca-Simona Horvath[2] and George Palamas[3]

Abstract

The process of architectural design aims at solving complex problems that have loosely defined formulations, no explicit basis for terminating the problem-solving activity, and where no ideal solution can be achieved. This means that design problems, as wicked problems, sit in a space between incompleteness and precision. Applying digital tools in general and artificial intelligence in particular to design problems will then mediate solution spaces between incompleteness and precision. In this paper, we present a study where we employed machine learning algorithms to generate conceptual architectural forms for site-specific regulations. We created an annotated dataset of single-family homes and used it to train a 3D Generative Adversarial Network that generated annotated point clouds complying with site constraints. Then, we presented the framework to 23 practitioners of architecture in an attempt to understand whether this framework could be a useful tool for early-stage design. We make a three-fold contribution: First, we share an annotated dataset of architecturally relevant 3D point clouds of single-family homes. Next, we present and share the code for a framework and the results from training the 3D generative neural network. Finally, we discuss machine learning and creative work, including how practitioners feel about the emergence of these tools as mediators between incompleteness and precision in architectural design.

Keywords

computational design, architecture, machine learning, design process, GNN, point cloud, generative design, artificial intelligence

[1]Center of Information Technology and Architecture, KADK, Kobenhavn, Denmark
[2]Department of Communication and Psychology, Aalborg University, Aalborg, Denmark
[3]Department of Architecture, Design and Media Technology, Aalborg University, Aalborg, Denmark

Corresponding author:
Panagiota Pouliou, Center of Information Technology and Architecture, KADK, Philip De Langes Allé 10, Kobenhavn 1435, Denmark.
Email: yotapouliou@gmail.com

Introduction

Artificial intelligence has gained widespread popularity in the last 4 or 5 years. While some of the algorithms used today were already available as early as the 1950s, there were periods when artificial intelligence (AI) research was more and less popular over the years. Periods of less popularity meant such severe funding cuts that they were called AI winters. There were two AI winters in the past: one in the mid-1970s and a second in the late 1980s (Leach N, 2022).[1] Today however, we can say that we are experiencing an AI summer: AI research has expanded significantly in all areas touched by technology, and it now seems hard to foresee an AI winter.

In architecture, AI has gained so much prominence that two books, partly sharing the same name: Architecture in the Age of Artificial Intelligence (Leach N, 2022; Bernstein, 2022),[1,2] were both published in 2022. Many of those studying computation and architecture have lately started using machine learning as part of their toolkit.

There are debates and different schools of thought working on different types of AI algorithms, however the ubiquity and availability of data that can be used to train machine learning algorithms is the reason for the current AI summer. In design, art, and architecture, generative models have gained a lot of attention. Generative models (Harshvardhan et al., 2020)[3] are machine learning models that can generate new data based on training data. Generative adversarial networks (GANs) (Goodfellow et al., 2014),[4] variational autoencoders (Doersch, 2016),[5] and diffusion models (Yang et al., 2022),[6] are all types of generative models.

There has been an explosion of AI-generated images based on text prompts since early 2022. Tools such as MidJourney (Midjourney, 2022)[7] and DALL-E 2 (OpenAI, 2022)[8] implement diffusion models to generate images based on massive databases of annotated images (Chaillou, 2022),[9] often owned and curated by big tech companies such as NVIDIA or Alphabet.

In the context of architecture, generative models that learn from annotated datasets have considerable potential, and a lot of valuable and intriguing work has come out of using text prompts to generate imagery (Del Campo et al., 2022).[10] Discussing new connections between language and architecture in the context of new tools, as done, for example, in Markus and Cameron (2002),[11] or Horvath (2022a)[12], Horvath (2022b)[13], Horvath et. al (2021)[14], takes new significance in this context. With that said, it is a necessity to investigate further how current AI-powered frameworks, specifically generative models, could impact architectural design in all its stages.

According to Rowe et al. (2017)[15], using digital tools in general for the design and production of architectural projects has made considerable contributions to design thinking in at least four areas. First, it has expanded the *exploration of conceptual and technical options* regarding new representational methods (e.g., renderings) and tools that provide higher precision (e.g., CAD drawings). Second, the *iterative power of generate-and-test procedures* can now result in generations of larger design spaces, and it may, therefore, assist in better problem-space structuring. Third, due to *broader access to information* or data, the evaluation and assessment of designs can be made with higher degrees of accuracy, scope, and technical sophistication. Finally, digital tools offer *better simulation techniques*. Wang et al. (2002)[16] described the architectural design process comparing the availability of tools in relation to the impact decisions have in the life of a building (see Figure 1).

The conceptual design stage is usually not formalized for most architects; however, the majority share similar incremental development over time from simple volumetric definitions (forms or 3D models) to gradually more detailed designs (Joyce, 2021).[17] Throughout all the design phases, 3D models are one of the most important data types used by architects in their design process. Machine learning-based generative tools for 3D models have been slightly less researched, although contributions were made, for example, by Koh (2022a)[18], Koh (2022b)[19], and dPrix et. al (2022).[20]

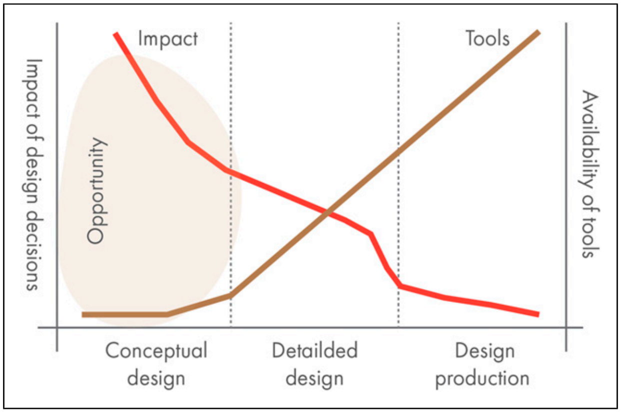

Figure 1. Tools that help in decision making for early stage design are fewer, although this early stage has the largest impact on an architecture project (image redrawn from).[16]

Building on previous work, in this paper, we aim to expand current exploration on the potentials and limitations of machine learning for architectural design using 3D point clouds and investigate how its use would influence design processes in the early design phases.

The rest of the paper is structured as follows: in the next section, we present a brief overview of machine learning in architecture and continue to focus on GANs, specifically 3D GANs. Next, we describe the materials and method used to conduct the study, followed by the main findings. We then discuss these findings and their possible implications for the field, focusing on how machine learning can influence creative work for architectural design.

Architectural design and machine learning

Technology has revolutionized not merely the way we design but also the way we think or imagine new designs (Al Qawasmi et al., 2007). There is extensive literature regarding architectural design accomplished by computational tools; however, there is ambiguity regarding a clear definition of computational design in architecture (Caetano et al., 2020).[21]

The application of digital technologies to architectural design has been the subject of research in several areas: computer-aided design (CAD) (Groover et al., 2007)[22], the use of integrated systems in architectural practice (Deutsch, 2007)[23], the role of virtual environments in design (Messinger et al., 2007; Whyte, 2007; Vite et al.),[24,2526] the impact of digital technologies on the built environment (Lou et al., 2022)[27], architectural education (Al Qawasmi et al., 2007; Gross and Do, 2003)[28,29]. The areas where digital technologies have changed architectural design workflows include: *conceptualization* (conceptual design of architecture but also the thinking processes these technologies mediate), *representation* (communication of the ideation phase) (Marble, 2012),[30] *realization* (the fabrication and manufacturing processes) (Wortmann et al., 2017)[31], and *evaluation* (the assessment and testing of architectural designs). Artificial intelligence can be used in any of these design phases, although the more technical and formalized the task, the easier it is to automate.

We focus on the first category, namely *conceptualization*, or the early stage conceptual phase of architectural design, the stage where decisions have the most impact (see Figure 1). Until now, most early-stage design tools have focused on environmental impact (Kharbanda et al., 2022[32]; Cavusoglu and Cagdas, 2018).[21]

On a theoretical level, many have wondered whether machines can be creative (Leach, 2022)[33] or whether they have the ability to dream (Del Campo et al., 2021).[34] Hansmeyer (2017)[32] suggested that we should *see* *machines as our muse, as a partner in design, or as a tool to expand our imagination*. Steinfield (2021)[35] categorized the use of machine learning tools in design, art, and architecture into three categories: machine learning as *actor* (models that co-design along with the designer), machine learning as *material* (generative models that provide new forms of design "material," usually curated by a designer), and machine learning as *provocateur* (models that provoke new models of thinking that later serve as sources of inspiration).

The increasing implementation of machine learning algorithms and pre-trained models has led Tamke et al. (2018)[36] to suggest that new architectural design practices should be based on ML approaches to better leverage data-rich environments and workflows. Several ML algorithms are already implemented in tools for architectural, civil, and environmental engineering (Belm et al., 2019).[37]

Initially, most machine learning applications focused on the analysis of real-life/existing data, but recently, machine learning algorithms are being applied to creative tasks as well (Belm et al., 2019).[37] For example, (As et al., 2018; Yang et al., 2019; Del Campo, 2020)[38,39] and[40] implemented deep neural networks to generate conceptual designs. Palamas (2022)[41] proposed an approach to creative knowledge mining using machine learning as a medium for guiding and motivating design exploration. Huang et al. (2018)[42] presented a study involving a two-step process in which architectural drawings were first recognized, and then new ones were generated. Algeciras-Rodriguez (2018)[43] utilized self-organizing maps (Miljkovi, 2017)[44] to produce hybrid forms that acquired characteristics from several input references. Dimensionality reduction tools were used for design data visualizations by Meng et al. (2020)[45] and Lunterova (2019)[46], while Harding (2016)[47] and Lunterova (2022)[48] used these for generative design exploration.

Generative Adversarial Networks

GANs were introduced in mid-2014 by Goodfellow et al. (2014).[4] Generative modelling (Goodfellow, 2016)[49] is an unsupervised machine learning task that involves distinguishing and learning complex data distributions of input data, that is, the training dataset (Van Den Oord et al., 2016),[50] to generate new samples that could have been part of the initial training dataset, meaning they have similar data distributions.

GANs consist of two sub-models: the *generator* model generates new samples, while the *discriminator* model aims to classify these samples as either real or fake (Goodfellow et al., 2020).[51] Typically, the generative network learns to map data points from a latent space—an embedding of a set of items within a topological space that locally resembles Euclidean space, where similar items are closer to each other—to a data distribution of a given training set, while the discriminative network distinguishes whether a sample is from the generative model distribution or the training dataset distribution. The two sub-models are trained together in a zero-sum game until the discriminator model cannot distinguish whether the sample that is evaluated is generated or pooled from the initial training set (Creswell et al., 2018).[52]

3D Generative Adversarial Neural Networks.

3D GANs have been used on a variety of problems, often using point clouds as input and/or output data. Applications of generative models have shown results in image-to-point-cloud transformation (Li et al., 2018),[53] text-to-voxel (Sanghi et al., 2022),[54] point cloud to point cloud completion (Zhang et al., 2021),[55] and point cloud up-sampling (Li et al., 2019).[56] These methods have achieved impressive results in computer vision applications.

Here, we investigate the capabilities of GANs in generating 3D point clouds from random latent codes. Latent codes provide generality and flexibility since they allow the model to assign arbitrary meanings to the items within the latent space. Even though GANs computing with 3D point clouds is still under-explored (Achlioptas et al., 2018),[57] there are several algorithms for training using point clouds.

The first to suggest and implement a method for point cloud generation were Achlioptas et al. (2018).[57] Raw point cloud GAN (r-GAN) was the initial model for generating point clouds from raw data points, and latent-space GAN (l-GAN) was a simplified version of r-GAN incorporating pre-trained autoencoders for pre-processing the data (Achlioptas et al., 2018).[57] Another GAN proposed by Valsesia et al.(2018)[58] used a dynamic graph convolutional network instead of a typical generator. Tree-GAN was proposed in 2019 by Shu et al. (2019)[59] and shaped a hierarchical structure in feature space using tree-structure graph convolutions. Finally, the most recent attempt at generating point clouds, a method called Controllable Point Cloud Generative Adversarial Network (CPCGAN) was introduced by Yang et al. (2021).[60]

Controllable point cloud generative adversarial network

CPCGAN not only performs better when it comes to the end results of the algorithm and in terms of its computational effectiveness compared to previous methods and it also allows the manipulation of the generated output in addition to generating segmented point clouds. A quantitative comparison of performance metrics of the models as proposed by Achlioptas et al. (2018),[57] and performed by Yang et al. (2021),[60] suggests that CPCGAN is the most effective algorithm to complete the task. CPCGAN (Yang et al., 2021)[60] generates point clouds from random latent codes by implementing a two-stage GAN framework.

The first network of CPCGAN is called *Structure GAN* while the second network is called *Final GAN*. Structure GAN learns the distribution of 32-point structure point clouds and outputs newly generated structure point clouds, along with their semantic labels. Subsequently, the output of the Structure GAN serves as an input for the Final GAN that learns the distribution of complete point clouds and can therefore populate the structure point clouds. Both networks implement typical generator models and PointNet-based (Qi et al., 2017)[61] discriminators. Yang et al. (2021)[60] have used the ShapeNet-Partseg dataset (Yi et al., 2016)[62] in order to showcase the effectiveness of CPCGAN.

Point cloud datasets

Even though there are many ways of storing three-dimensional and spatial information, most 3D GANs use point clouds. A point cloud is a set of data points in a three-dimensional coordinate system, defined by X, Y, and Z coordinates (Hana et al., 2018).[63] Besides the coordinate values, the dataset may contain other features and attributes, depending on its creation processes, such as reflection intensities or RGB color values.

The reason point clouds are popular is the simplicity of their components. Using single points, with no attributes of scale, rotation, etc., can be handled and computed much easier in a large amount (Horvath, 2014).[64] Additionally, as the outputs for environmental scanning technologies such as LiDar or Kinect are point clouds, this data type has become more frequent (Wandinger, 2005).[65]

Several research projects in the past developed new approaches toward collecting or generating segmented point clouds. Segmentation of a point cloud refers to classifying the points into multiple homogeneous regions. The points belonging to the same region (for example, all points representing windows) will have this meta-data attached to them. Most of these studies focused on advancing the field of Computer Vision in different ways. For example, The KITTY Vision Benchmark (Geiger et al., 2012)[66] was created to assist autonomous driving by implementing urban-scale spatial information such as cars, trees, roads, pedestrian streets, and building blocks. The ModelNet dataset (Wu et al., 2015)[67] includes indoor space data, mostly focusing on objects and furniture detection. The ShapeNet-Partseg (Yi et al., 2016)[62] consists of 16 object classes, each segmented according to its parts (e.g., airplane: tail, body, wheels, wings). The ArCH dataset (Malinverni et al., 2019)[68] consists of 17 large-scale heritage-buildings annotated point clouds. Finally Croce et al. (2021)[69] presented a semi-automated way of labelling heritage buildings and provided a dataset of 16 annotated point clouds of heritage buildings.

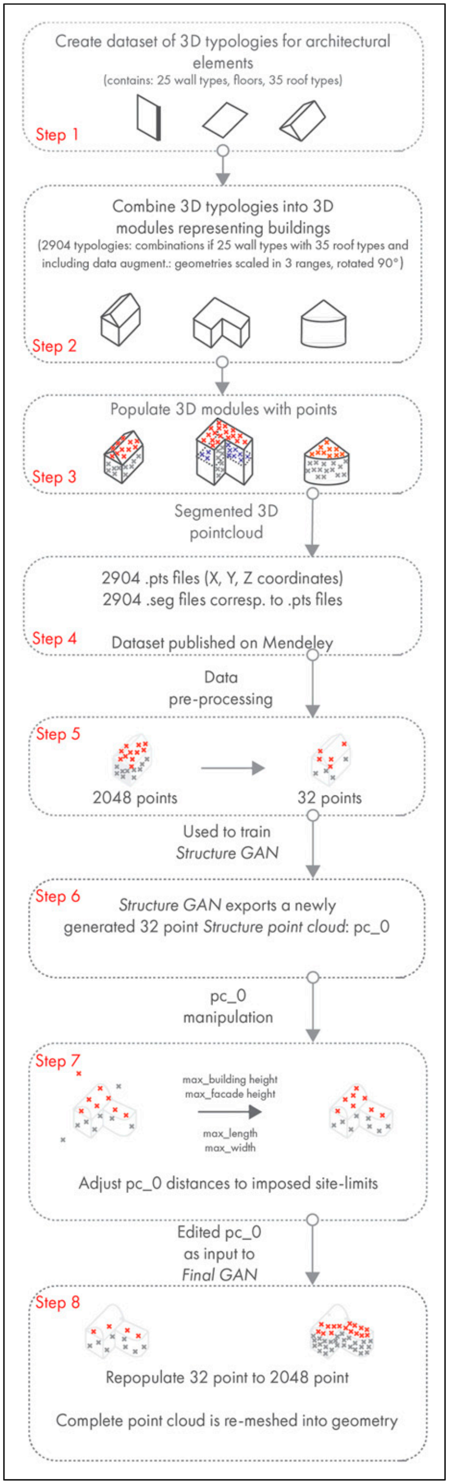
Create dataset of 3D typologies for architectural elements
(contains: 25 wall types, floors, 35 roof types)
Step 1
Combine 3D typologies into 3D modules representing buildings
(2904 typologies: combinations if 25 wall types with 35 roof types and including data augment.: geometries scaled in 3 ranges, rotated 90°)
Step 2
Populate 3D modules with points
Step 3
Segmented 3D pointcloud
2904 .pts files (X, Y, Z coordinates)
2904 .seg files corresp. to .pts files
Step 4
Dataset published on Mendeley
Data pre-processing
Step 5
2048 points
32 points
Used to train Structure GAN
Step 6
Structure GAN exports a newly generated 32 point Structure point cloud: pc_0
pc_0 manipulation
Step 7
max_building height
max_facade height
max_length
max_width
Adjust pc_0 distances to imposed site-limits
Edited pc_0 as input to Final GAN
Step 8
Repopulate 32 point to 2048 point
Complete point cloud is re-meshed into geometry

Figure 2. Framework for generating the dataset and training the CPCGAN algorithm.

Speculative hybrids: a framework for generating conceptual architectural forms

We implemented a 3D GAN trained with a dataset of building geometries represented as annotated point clouds. We call this framework and its results *Speculative Hybrids*. Figure 2 shows the framework with its eight steps, and below, we describe each of these steps in detail.

After implementing the framework, we presented the results of one instantiation to 23 practitioners of architecture and surveyed them to understand if and how this framework could assist in practice and how practitioners see creative work being influenced by technology in general and machine learning tools in particular. In this section, we also describe the participants and give an overview of the questions from the survey.

Dataset creation (steps 1–4)

At the moment of writing, to our knowledge, there is a scarcity of datasets of segmented building geometries. We created a simple dataset using a semi-automated process and used it as input for the CPCGAN algorithm. The dataset could be representative of single-family houses with different typologies. We created the dataset in two stages: first, we created module geometries that represent building components in Rhinoceros3D (McNeel, 2019),[70] then we generated point clouds based on these geometries using the Cockroach plug-in (Vestartas, 2020)[71] for Grasshopper.

Module geometries (step 1&2). To generate the dataset, we used Rhinoceros3D to create a library of modules representing typologies for three building elements: walls, roofs, and floors represented as surfaces and placed on separate layers. We created 25 wall typologies to which we applied data augmentation methods to increase the size of the dataset. In more detail, we mapped the walls to three different scales and rotated them at 90°. Some of the plan variations according to the wall modules can be seen in Figure 3. Next, we created 35 different roof typologies and combined each roof typology with each wall module, respectively (see Figure 4). Apart from the wall and roof typologies, we also generated floors as simple surfaces.

Figure 5 shows some of the generated building geometries. Next, all surfaces representing a typology were joined into one polysurface, converted into a mesh, and saved to a new corresponding layer. The process was repeated for all three module categories. Its outcome served as input for the generation of point clouds.

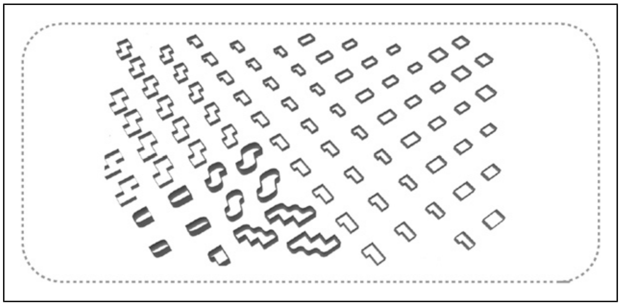

Figure 3. Step 1: The 25 wall typologies generated in the first step of the framework. Each of these modules was combined with each roof typology to create the building geometries.

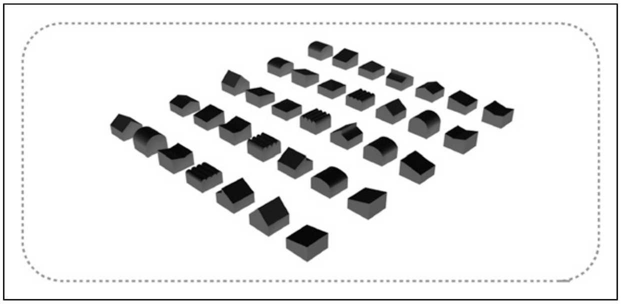

Figure 4. Step 1: The 35 roof typologies generated in the first step of the framework. Each roof module was combined with each wall typology to create the building geometries.

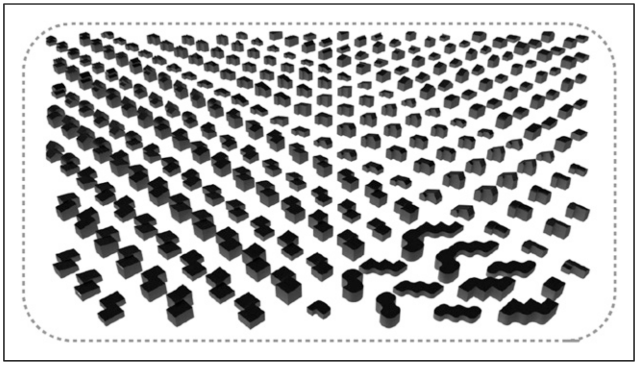

Figure 5. Step 2: The complete array consists of 2.904 simple building geometries that were transformed into point clouds for the training of CPCGAN. Here, diverse building generations are showcased.

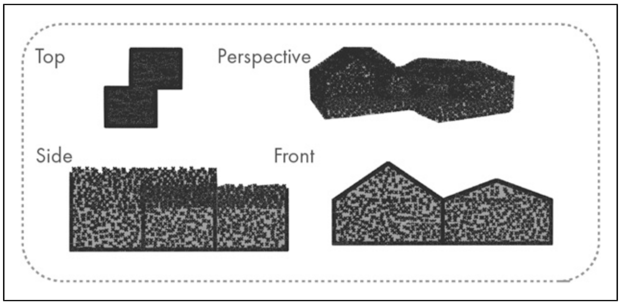

Figure 6. Step 3: Example of a dense point cloud that was generated by populating the meshes with points using the Cockroach plug-in.

Point cloud generation (step 3). These meshes served as a basis for the point cloud generation: we used the PopulateMesh node from the Cockroach plug-in (Vestartas, 2020)[71] for Grasshopper. PopulateMesh takes as input a mesh, a number of points to populate the geometry with, and a sampling type. We used Poisson Disk Sampling (Bridson, 2007),[72] which helped to provide a more uniform distribution of sample points along the converted meshes. We used 1200 points for the wall meshes, 1,000 for the roof meshes, and 400 for the floor meshes. The function's output is a point cloud for each building component, which was then merged into a unified point cloud, as shown in Figure 6.

Following the generation of the points of each module, we created a list of the labels corresponding to the class to which each point belongs (i.e., wall = 1, roof = 2, and floor = 3). The lists for the different components of a building were then merged into a single list. Similarly, the segmentation lists were merged into a single list with the labels of each point of the point cloud.

Training CPCGAN and generating new samples (steps 5–8)

The implementation of the CPCGAN algorithm was done in three steps: the first step involved pre-processing the data, the second step consisted of training the CPCGAN algorithm, and finally, generating a random sample or the generation was controlled.

Pre-processing the data (Step 5). We pre-processed the data to fit the implementation of CPCGAN, which required the point clouds to be no larger than 2,048 points. We used a script that randomly sampled 2048 points per "house" for implementing the training. Consequently, a *samples folder* was created where the chosen points were saved into a single file for each point cloud. Additionally, we processed the sampled data and created a structure point cloud for each point cloud in the dataset. The structure point cloud consisted of 32 points and was created using the Growing Neural Gas (GNG) algorithm (Chevi, 2020).[73] Growing Neural Gas is a clustering algorithm introduced by Bernd Fritzke. Clustering is the process of organizing a collection of k-dimensional vectors into groups whose members share similar features (Holmstrm, 2002).[74]

In the original implementation of CPCGAN by (Yang et al. (2021),[60] K-means (one of the most commonly used clustering algorithms) was used. We chose to implement GNG to achieve a higher performance rate in the creation of the structure point clouds, aiming for better topology preservation. While K-means clustering may be faster, according to Daszykowski et al. (2002),[75] Growing Neural Gas provides better results.

Training CPCGAN (Step 6). We trained CPCGAN using an experimental framework called DLNest (Yang, 2022).[76] This framework allows the training and automatic loading of machine learning models. In our study, we created three models: the first model had 10 epochs, the second model had 500 epochs, and the third model had 2600 epochs. The duration of the training for each model was approximately 20 min, 30 h, and 96 h, respectively. We present in the following sections the results of the last training.

Generating samples and controlling the generation (Step 7&8). New point clouds were generated after training the two GANs of CPCGAN. Using DLNest, *Structure GAN* generates first, a 32-point structure point cloud, and later, *Final GAN* generates a 2048-point fully populated point cloud.

To control the generation, we created a function that mapped the newly generated point cloud to a hypothetical site and its hypothetical building regulations: percentage of site coverage, maximum facade height, and maximum building/roof height (see Figure 2). This was an intermediary step between the two sub-models of CPCGAN, namely *Structure GAN* and *Final GAN*. All the points of the generated *Structure point cloud* were set to cover a maximum *width* and *length* on the plot, and to therefore comply with a maximum site coverage. For the points representing walls, a maximum height limit was set in order for the generation to respect the maximum facade height allowed. The points representing the roof were also limited to a maximum height limit in order to follow a rule for the maximum total height of the building. The hypothetical site was 2,460 m^2, and the regulations we imposed on it are shown in Table 1. The *width* and *length* parameters were set to 10 m and 12 m, representing a building percentage of 5% of the total site area.

Qualitative evaluation of the framework

After creating the dataset and training CPCGAN, we presented the framework together with a survey to 23 participants, all practicing architects. In individual sessions (either online or in-person), we described the *Speculative Hybrids* framework and introduced them to a case study of the framework in use. We generated 10 building solutions according to the hypothetical site presented. The participants were shown graphical representations of the generated volumes and asked to answer an online survey that collected demographic data, data about work experience, and asked open-ended questions on the use of computational design tools, their limitations, and possible ethical considerations related to machine learning for the profession. The graphical input shown to the participants can be seen in Figure 7, and represents the top and perspective view of the generated volumes. Given that the framework did not have an interface yet, participants did not interact with it themselves.

We aimed to evaluate the potential and limitations of generative machine learning tools during the early stages of design and to see whether it would be useful to incorporate this framework into a tool such as a Grasshopper or Dynamo plug-in. Additionally, we briefed the participants on the frameworks' potential to be

Table 1. The building regulations of the site that the framework was calibrated to generate volumes for.

Building regulation	Max value
Site coverage percentage	5%
Width	10 m
Length	12 m
Facade height	8 m
Total height	15 m

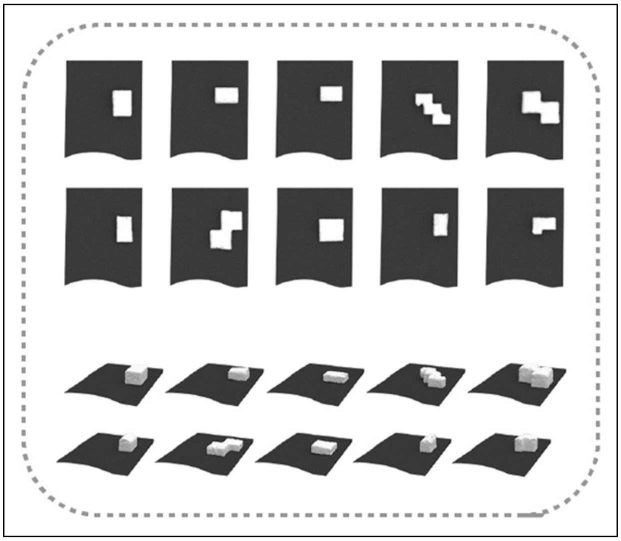

Figure 7. The input shown to the participants, as printed material, together with the survey, in order to evaluate the *Speculative hybrids* framework.

trained with different geometries and architectural styles, given adequate datasets, either created (by themselves) or readily available from others.

Participants. A total of 23 individuals participated in the evaluation, with one being under 25, twenty between 25 and 35, and two between 36 and 45. Their educational backgrounds and specialities ranged from master's students to PhD candidates or practitioners in architecture, computational design, landscape architecture, and architectural engineering. We selected practitioners of varying backgrounds and ages without regard to any other criteria aiming for responses from a broad range of practitioners.

Findings

The findings of our study are twofold: on the one hand, we report on the use of the CPCGAN algorithm for generating 3D models of buildings, and on the other hand, we present the results of the survey where we interrogated practitioners of architecture with questions about the potential use of machine learning tools for the conceptual stages of architectural design.

Generated building volumetries (Step 8)

CPCGAN was used to generate 32 initial point cloud structures. Following this, a 2048-point structure point cloud was generated. Figure 8 illustrates an example of each step. Next, we created meshes from the 2048-point point clouds, see Figure 9. Finally, Figure 10 shows a closer look at the input and output data from the trained CPCGAN model. The output data significantly resembles the input data; however, it is not completely identical to it. By inspecting the geometries of the generated buildings, we confirmed that these comply with the site-specific regulations introduced in the previous subsection. On the contrary, the same does not apply to all the input samples used to train the network. This was achieved by controlling the output of the Structure GAN before it was fed to the Final GAN.

Practitioners' perspectives on the framework and the use of machine learning in conceptual architectural design

We initially asked participants about the tools they use during the early design and ideation stages. They mentioned a mix of digital and analogue tools, including hand-drawn sketches, physical modelling (foam,

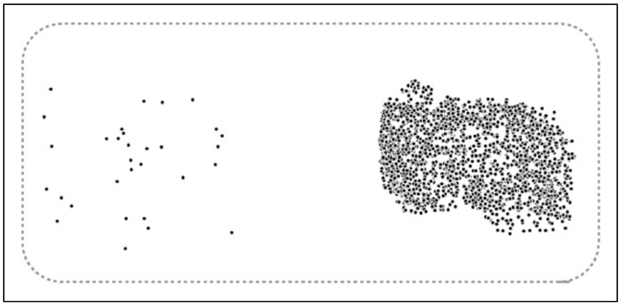

Figure 8. Step 8: An example of a generated *Structure point cloud* can be seen on the left. The right side of the image shows a fully populated point cloud with 2,048 points. The generated point cloud is able to reproduce the global structure of the original point cloud, as well as the local details.

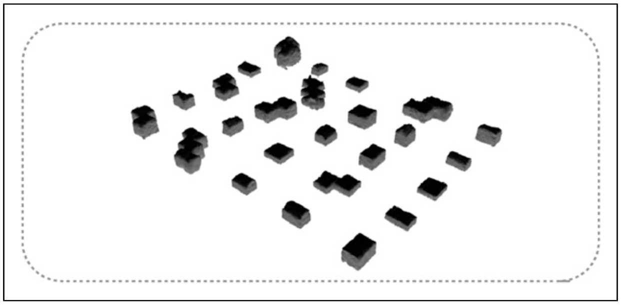

Figure 9. Step 8: Perspective view of the generated *Speculative Hybrids* after the resulting point cloud were re-meshed into geometries.

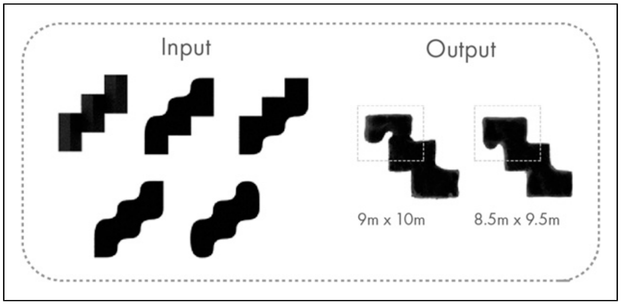

Figure 10. Step 8: A closer look at the input and output data from the trained CPCGAN model.

cardboard, or clay models), digital drawing tools, such as vector raster graphics software, and 3D modelling using both BIM software and free-form modeling software. Only one participant reported using machine learning models and computer vision in the early ideation phase. Most participants said that the design process begins with conceptual sketches on paper, followed by tests on how accurate the sketches were in terms of scale or proportion using digital software tools. Some participants mentioned using algorithms to calculate building performance (e.g., in terms of light or wind) by implementing various scripts.

On the Speculative Hybrids framework. We then asked participants whether they could find the *Speculative Hybrids* framework useful in their early-stage design. The vast majority stated that they could see themselves working with such a tool and they looked forward to its further development. One participant stated that *the method should be implemented in a plugin or as an online tool*, while two others mentioned that it should function as a browser-based tool to serve as a platform to communicate between the architect and their clients. Two of the participants stated that the "artistic" or "creative" parts should be left to the designers: *I think it's a good and useful thing as long as it doesn't take a too large part of the creative work* and *I believe that any help is welcome in the design process as long as the artistic part is assured to us people*. They mentioned that more interaction with the data would be useful (including interacting with the dataset used to train the tool). Additionally, one of the participants mentioned that they would like to have more information regarding added future functionalities to the tool, while another one questioned whether such a tool is directed only to practitioners who specialize in AI or computational design for architecture.

On potentials and limitations of computational and machine learning tools for architectural design. Most participants stated that *technology is the future of the profession* and that the architectural practice is facing higher demand in terms of being able to produce outcomes that are predictable (i.e. we can know how buildings will be used, how they will behave from an energy consumption perspective) and whose performance can be measured. Therefore computational design and even machine learning are expected to assist architects to design these more predictable solutions. Tools are expected to support material use optimization, efficient management of architectural projects at all stages (from design to construction and post-occupancy) as well as technology assisting in coming up with sustainable design options. Participants also stated that computational design gives the opportunity to generate and handle large amounts of information, numerous editions of an idea, and complex geometries. However, according to our particioants, aspects like the connection with the surroundings, view and orientation, aesthetics, and ergonomics are all decisions that should be taken outside of an algorithm.

Many participants noted a series of limitations they see in current computational design tools. One mentioned that *(the tools don't allow) very good communication between mind and hand and [...] there is less connection between the idea and one's senses*, while another mentioned that *[tools] lack the flexibility of the non-software prototyping*. Another participant stated that *using technology is sometimes more complicated than doing the design by yourself*, while one mentioned that *[tools] force you to think like a technologist rather than to think like a creative*. Limitations are not only linked to the creative process itself but also to how the tools frame architectural design problems: *some tools are too strictly framed or without any socio-spatial considerations* or try to *encode architectural qualities into quantitative metrics without proper research*. Apart from limitations to the creative process, and the problem space formulation, or the reduction of complex qualitative problems to numerical problems, one participant mentioned the high cost of computational tools: *high fidelity—especially in the field of performance and simulation studies—often leads to computationally intensive workflows, and therefore results in higher costs and processing time. The question is how to overcome the tie of the generative track to fast, but inaccurate lightweight types of computation and the analytical track to heavyweight but slow types, when complex architectural problems require generative approaches and a general demand for more frequent and better feedback design cycles*. Finally, others mention the current tools' lack of interaction capabilities, and how, quite simply, they are *not tailored to the architectural design workflow.*

Ethical and sustainability implications of using computational and machine learning tools for architectural design. Contrary to our expectations, only two participants said they found no ethical implications to using computational design tools for architecture. The other 21 expressed a variety of concerns. For example, one stated that *you cannot map humans and transform their needs and well-being into an algorithm* or *[the tools assume] the reduction of human experience into parameters*. These concerns relate to oversimplifying the design process (reducing complex design problems to "simple" algorithmic problems): *many generative design processes rely mostly on mere typological or performative criteria and exclude human-scale, social responsiveness, or contextual consideration, all features that are conventionally regarded as being central to design*. Others mentioned the makers of technologies and how their ways of thinking may influence what designers using those technologies can afford: *potential bias from the creator of the tool implanted in the tool itself* or *we are only as good as the dataset* or more *the idea is never something created to fit the needs of the place/person/project, is something randomly generated based on software, so many projects could possibly be almost identical and applicable everywhere (not that avoiding the use of the software would prevent that, because it is already happening, but maybe the use of the software is accentuating it)*. Two of the participants expressed worries regarding the future of the profession itself: one stated that architects might be made irrelevant by new generative tools, such as the framework we presented to them, while another mentioned that we might not need structural engineers in the future. Moreover, two participants commented on the social inequalities related to who has access to both the tools themselves (including access to expensive software or infrastructures of robotic fabrication), but also regarding access to education about how to use computational design tools.

To summarize, all practitioners agreed that technology is the future of the profession and upon the immensely positive impact it has had on architecture over the last three decades. Consequently, most were enthusiastic about the *Speculative Hybrids* framework and stated they looked forward to seeing it as a tool they could interact with directly. They felt they should be able to interact with both the parameters that frame the site constraints and the dataset used to train the model. However, most participants also expressed concerns regarding computational design and machine learning tools. Their critiques revealed concerns regarding computational tools in general; however, they can also relate to the *Speculative hybrids* framework itself. These concerns can fit into two broad categories: (1) Concerns related to design processes in general and (2) Ethical or sustainability issues. Below, we summarize these main concerns:

- **design-process related:** current tools are too rigid and *force designers to think as "technologists"*;
- **design-process related:** tools *prolong the distance between an idea (in mind) and its representation (on paper/screen).* In short—the issue of mediation is both too complicated (it is hard to express an idea), and too limiting (only certain elements are available, therefore, certain ideas are easier to express);
- **design-process related:** building on the above, *large datasets do not necessarily mean better design workflows*: there is a need to understand data granularity over space and time, and work with what is relevant in specific cases;
- **design-process related, and ethics related:** *computational design tools force complex design problems into numerical formats, that do not translate one-to-one.* While some aspects of architectural design can be translated to numerical problems (i.e., volumes, square meters, light, etc.), the *human, social and well-being factors might not be directly transferable to parameters that can feed algorithms*;
- **ethics and sustainability-related:** tools, especially those that are based on vast amounts of data, require a lot of *computational resources, which makes them energy-expensive*;
- **ethics and sustainability-related:** *accessibility challenges*—there are still only a few architecture schools around the world that teach computational design to architecture students, or that have robotic fabrication laboratories. Additionally, software licenses are inaccessible to many small design companies and even universities around the world. This contributes to and deepens inequality across the profession;

Discussion

As computational design tools in general, and machine-learning-powered (ML-powered) frameworks in particular, enter architectural design practices, it is relevant to investigate how this is changing design processes and how architects perceive this ongoing and constant retooling of a profession that spent many centuries using relatively unchanged tools (drawing, perspective, axonometry, and physical modelling).

The history of architecture shows how the architectural profession broke down into smaller parts: the master builder became the architect and the structural engineer, and later, the industrial revolution further split the field into different types of engineers and subcontractors. This was necessary as basic living needs evolved (i.e., water or electricity became the baseline for habitable spaces, and fire safety became part of building regulations). It was accomplished by a *rationalization* (Chaillou, 2022)[9] or systematization of the profession of architecture. A change that arguably started as early as the Renaissance but gained momentum during the first industrial revolution and modernism with its Bauhaus movement. But as the practitioners make clear in their comments, and as (Bernstein, 2022)[2] argues, *[the profession] involves too much ambiguity, need for* *judgment and trade-offs and demand to solve wicked problems at a variety of scales* for any worries about machine learning tools fully replacing most of the tasks architects do. It is more likely, as (Bernstein, 2022)[2] argues, that algorithms will automate some *technical* tasks.

Many practitioners stated they favor analogue techniques for initiating their conceptual design phase. This fact may be due to perceived and existing limitations of computational design tools (*they encode architectural qualities into quantitative metrics*), because of their inaccessibility (*computational design requires skills that not all of us have*), or due to the cost of their implementation (*using technology is sometimes more complicated than doing the [initial] design by yourself*). The ones that use computational design tools utilize them to simulate how a building will perform and try to fit environmental sustainability metrics using, for example, Ladybug (Roudsari et al., 2013)[77] or LearnCarbon (Kharbanda et al., 2022).[78] Most participants noted that using computational design tools involves forcing oneself to think in ways that are too rigid or limiting and that computational tools involve no socio-spatial considerations. These early stages of architectural design include decisions that, as one of our participants mentioned *need to be taken out of the algorithm*.

In this section, we discuss our findings around two pillars. First, we place machine learning tools for early-stage design work in a space between incompleteness and precision—as per the definition of Rowe et. al, (2018).[15] Next, we discuss the notion of latent space that machine learning algorithms work with as a conceptual space where abstract, compressed ideas are stored and discuss it concerning creativity as defined by Margaret Boden.

On the speculation: navigating a space between incompleteness and precision in early stage architectural design

Despite exhibiting a significant variety, all the outputs of the training of the 3D GAN belong to the same building typology as those used to train the model. Being limited to an artificially generated dataset, they are less detailed than real-life buildings. A sufficiently large collection of segmented point clouds of realistic buildings would have allowed us to explore even further the potential of this framework. The CPCGAN algorithm was slightly altered to improve its performance by implementing the GNG algorithm to create the structure point clouds preserving the same proportion between the number of points for each label as in the original point cloud. We created point clouds using Poisson Disc Sampling, and therefore, they have no density differentiation regarding each category. However, the use of the proposed framework would achieve significant results if applied to point clouds deriving from scanning technologies, since these would most likely present heterogeneous densities.

Understanding the usefulness and limitations of machine learning algorithms for architectural design builds on understanding and defining design thinking and the designers' way of knowing and doing (Schön, 2017).[79]

According to Schön (2017),[79] design thinking constitutes of the following: a process of *reception* (perception), *reflection* (interpretation), and *reaction* (transformation). Building on this, Oxman (2006)[80] defined four major areas describing the process of design: *problem formulation*, *synthesis/generation*, *representation*, and *evaluation*. Machine learning could intervene in any of these stages, but these tools can be useful or limiting at each stage is different. Rowe et al. (2018)[15] summarized the understanding of design thinking in architecture and suggested its characterization by the properties of *incompleteness*—what Schön describes as design thinking (reception, reflection, and reaction) and *precision*—the evaluation and functionality of the results of design thinking. Therefore, according to Rowe et al. (2018)[15], design thinking is the process that exists between the two, aiming to balance for continued problem-space structuring (see Figure 11). Building on Schön's work, Bernstein (2022)[2] divides the tasks of an architecture professional on a continuum from more specific and technical, thus easy to automate, to complex, ambiguous, and requiring

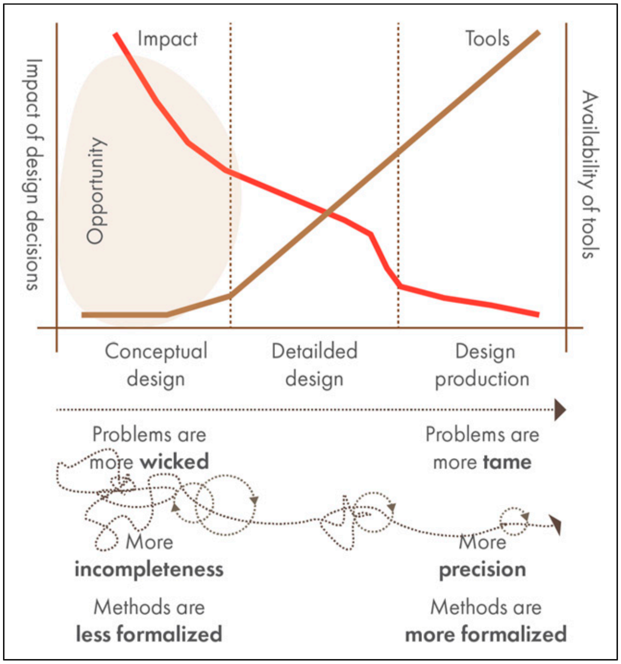

Figure 11. In the early stages, design methods are less formalized, problems are more complex, or wicked, and designers deal with more incompleteness. As the project evolves, all the stakeholders involved in the design process make design decisions that help disentangle the complex problems faced in the beginning.

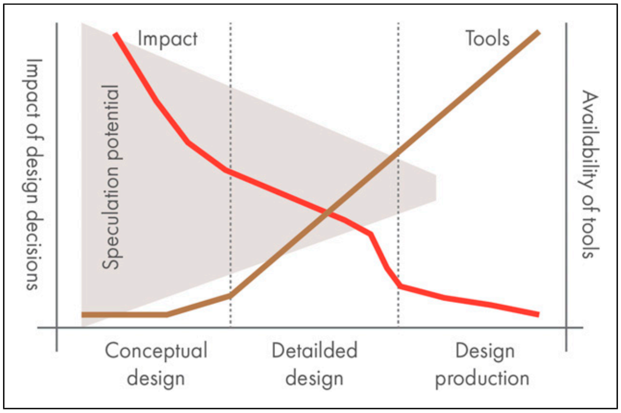

Figure 12. The room, need, and potential for provoking speculation to sit in the less structured, more wicked earlier design stages.

moral or ethical judgments. This continuum goes between (1) procedural, (2) procedural to integrative, (3) integrative, and (4) integrative to perceptive.

The early-design stages are also where there is more room for speculation—what we understand as creativity, or what one of our participants called the “artistic part” that should remain with “us humans” (see Figure 12). In Bernstein’s continuum, this space would be integrative and perceptive. Looking at the responses of our participants, it seems that ML-powered tools for the early-stage design should function as provocateurs (or points of departure) for new designs, rather than as actors (partners and co-creators) or materials (the designers create and use the tools similarly to craftsmanship), in Kyle Steinfeld's categorization (Steinfeld, 2021).[35]

Some generated outputs presented features not included in the initial dataset (e.g., differentiated curves, see Figure 10). As a fact, we can argue that the framework produced unexpected results (incompleteness) that fit within the regulations we imposed on them (precision). These results came from the interplay between the two GANs used in the framework.

Zooming out from our specific case, the question becomes: how should ML-powered frameworks be calibrated to foster speculation in the early design stages and to help provoke new ideas? Provocation was well defined as part of critical design studies (Dunne and Raby, 2013).[81] Provocative design refers to design approaches that operate in a design space where asking questions is as important as solving problems (Ozkaramanli and Desmet, 2016).[82] But provocation is always bounded by a time, a space, and its receivers who should be able to interpret it. Using machine learning as a provocateur can take different avenues as we see them. First, the tools can serve as a starting point or inspiration that then is continued by the designer and detailed or morphed into a concept. Second, mistakes done while using machine learning frameworks could be honed and used as starting points for design. Third, the tools can serve as provocateurs for the designer. Provocation here is understood as more than inspiration or a starting point and could help with perspective-taking and triggering personal dilemmas. This fact could assist in creating productive speculations in what Marenko (2018)[83] calls future crafting, as a necessary condition for any early-stage design. Finally, the tools could help the designers create provocative projects for the receivers (i.e. their clients) that would foster reflection, critical engagement, and debate. This future crafting done with machine learning tools sits in a space between incompleteness and precision: on the one hand, as our participants stated—ML tools can be used to simulate the future (e.g., through environmental simulation tools). On the other hand, they can be used as inspiration or a provocation for imagining possible futures.

On the hybrid as a way of seeing in latent space

In generative adversarial neural networks, latent space is the intermediate layer between noise and a generated output (be it an image, a point cloud, or other data types). The latent space stores a compressed and simplified representation of the data used to train the neural network and could be imagined as an n-dimensional model (Chaillou, 2022).[9] A walk in latent space could be imagined as a walk between two of these compressed data points. If we oversimplify and imagine we had a latent space containing only two points: one represented by a cube, and another by a sphere—a walk in latent space between these two points would take all the intermediate steps between the cube and the sphere. This hybrid is an instance of a blend and is the way of seeing of these generative adversarial neural networks. One could argue that all outputs of generative models are hybrids of this type, and while ML-frameworks are excellent at generating vast amounts of hybrids, they are also limited by their way of seeing when it comes to creative expression.

According to Margaret Boden, there are three types of creativity.[84,85] *Combinational* creativity refers to when new ideas come from making unfamiliar combinations of familiar ideas (poetic imagery, collage). This type of creativity is where machine learning models excel, as a walk in latent space would create a large number of hybrids interpolating between different points. *Exploratory* creativity refers to conceptual spaces that are structured styles of thought (artistic movements such as modernism, and baroque). A new idea within that style of thought is considered “exploratory creativity.” Designers and programmers who come up with new algorithms and ways of generating datasets could be said to exhibit exploratory creativity within ML-powered design. They could be similar to those using machine learning as a material in their design processes.

Finally, when working within a conceptual thought style (such as ML-powered design), understanding that the thought style has limitations and coming up with ideas that transcend the thought style is what Boden calls *transformational* creativity.

While the first two types of creativity would require staying within the algorithm, or staying within a thinking style that allows one to create and train new algorithms (thus working with precision and automation), transformational creativity would require one to take decisions outside of it. Engaging with these tools and using them as starting points or as provocateurs could help spark critical discussions about their limitations, and could provoke new ways of thinking that are more speculative and imprecise.

However, as our participants also note—it is momentous to be especially critical about automation in tasks that involve data about people and in tasks that involve qualitative sensibilities. It is also important to take into consideration the environmental aspects of more data: where is it stored, for how long, by whom, and how much energy is needed to store and process it.

Conclusions

In this paper, we report on our efforts to understand how machine learning can influence early-stage conceptual design for architecture. We present a framework that allows the exploration of building volumetries based on site-specific regulations. We trained the framework using a dataset of single-house buildings represented as annotated point clouds. We generated the dataset and made it available for the research community along with the code to implement and modify the framework for future use. By presenting the framework and results from experimenting with it to architecture practitioners, we learned that this framework could be able, at a certain level, to inspire architectural practitioners in the ideation phase through exploration, within the context of site-specific regulations. According to our participants, ML-powered tools in early-stage architectural design should function as inspiration (starting points) or provocation for future crafting, where decisions are taken both outside the ways of seeing the algorithm and outside the ways of thinking these algorithms impose if one should use them as tools for design. Without allowing themselves to get outside of the “algorithm,” designers cannot be sufficiently critical about possible ethical or sustainability issues related to using these tools.

Several investigation avenues emerge as a natural continuation of this project. The dataset used to train CPCGAN is relatively simple, describing different scales of single-family houses. Despite its simplicity, generating such a dataset is a rather time-consuming process which is why we make the dataset publicly accessible to the research community[86] together with the code[87] we used.

Generative algorithms could be implemented to automate and expand the dataset we created. Additionally, we would like to explore the possibility of using real-life data for the same process. This could be achieved with the use of 3D scanning technologies like LiDar cameras or through a collaboration with an architectural practice, to train CPCGAN with point clouds generated from their own BIM models.

Regarding the performance of CPCGAN, we would like to employ the Growing Neural Gas algorithm for downsizing the point clouds to 2,048 points, a process that happens randomly in CPCGAN in its current implementation. Additionally, the proposed method for the generation of architectural forms was not used by the survey participants. The method requires, in its current form, high computational power and does not have an interface for field practitioners. Therefore, a logical development would be to implement the proposed method in a ready-to-use Rhinoceros3D plugin that builds on a pre-trained algorithm.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received financial support for the research from the Human Centered AI cluster, Department of Communication and Psychology, Faculty of Humanities and Social Sciences, at Aalborg University.

ORCID iDs

Panayota Pouliou https://orcid.org/0000-0002-8069-3382
Anca-Simona Simona Horvath https://orcid.org/0000-0001-5371-5657

References

1. Leach N. *Architecture in the age of artificial intelligence: an introduction to AI for architects*. London, UK: Bloomsburry Publishing, 2022.
2. Bernstein P. *Machine learning: architecture in the age of artificial intelligence*. Abingdon, Oxfordshire: Routledge, 2022.
3. Harshvardhan GM, Gourisaria MK, Pandey M, et al. A comprehensive survey and analysis of generative models in machine learning. *Computer Science Review* 2020; 38: 100285.
4. Goodfellow Ian J, Jean PA, Mehdi M, et al. Generative adversarial nets. *Proceedings of the 27th International Conference on Neural Information Processing Systems* 2014; 2: 2672–2680.
5. Doersch C. Tutorial on variational autoencoders. Online, arXiv preprint, https://doi.org/10.48550/arXiv.1606.05908, 2016
6. Yang L, Zhang Z, Song Y, et al. *Diffusion models: a comprehensive survey of methods and applications*. arXiv preprint, 2022
7. Midjourney. https://www.midjourney.com/, 2022.
8. OpenAI. DALLE 2. https://openai.com/dall-e-2/, 2019.
9. Chaillou S. *Artificial intelligence and architecture: from research to practice*. Basel: De Gruyter: Birkhaäser, 2022.
10. Del Campo M and Leach N (eds). *Machine Hallucinations: Architecture and Artificial Intelligence*. Hoboken, New Jersey: John Wiley & Sons, 2022.
11. Markus TA and Cameron D. *The words between the spaces: buildings and language*. Abingdon, Oxfordshire: Routledge, 2002.
12. Horvath AS. How we talk(ed) about it: Ways of speaking about computational architecture. *International Journal of Architectural Computing* 2022; 20(2): 150–175. https://doi.org/10.1177/14780771211070006
13. Horvath A-S. ComPara: A Corpus Linguistics in English of Computation in Architecture Dataset. *Data Brief* 2022; 42: 108169, https://doi.org/10.1016/j.dib.2022.108169.
14. Horvath A-S, Vite C, Møller NLH and NeffmessyBIM G. Augmenting a building information model with messy talk to improve a buildings' design process. In M Menéndez-Blanco, S Uğur Yavuz, J Schubert, D Fogli, F Paternò and editors, CHItaly 2021 Joint Proceedings of Interactive Experiences and Doctoral Consortium. Vol. 2892. CEUR Workshop Proceedings. 2021. p. 7-14. 2
15. Rowe PG, Mohsen M, Bechthold M, et al. *Design thinking in the digital age*. Stenberg Press: Harvard University. Graduate School Of Design, 2018.
16. Wang L, Shen W, Xie H, et al. Collaborative conceptual design—state of the art and future trends. *Computer-aided Design* 2002; 34(13): 981–996.
17. Joyce SC. AI as a collaborator in the early stage of the design. *The Routledge companion of Artificial Intelligence in Architecture. Imdat As and Basu Prithwish (eds)*. Abingdon, Oxfordshire: Routledge, 2021.
18. Koh I, 2022. Architectural Plasticity: The Aesthetics of Neural Sampling. Wiley Online Library. *Architectural Design*, 92(3), pp. 86-93.
19. Koh I. In: JS Gero (ed) *Voxel Substitutional Sampling. Generative Machine Learning for Architectural Design*. New York City: Springer International Publishing, 2022, pp. 379–398.

20. dPrix W, Schmidbaur K and Bolojan D, et al. The legacy sketch machine: from artificial to architectural intelligence. *Architectural Design* 2022; 92: 14–21.
21. Çavuşoğlu ÖH, Çağdaş G and Leitão A. Computational design in architecture: Defining parametric, generative, and algorithmic design. *Frontiers of Architectural Research* 2020; 9(2): 287–300.
22. Groover M and Zimmers EWJR. *CAD/CAM: computer-aided design and manufacturing*. London, UK: Pearson Education, 1983.
23. Deutsch R. *BIM and integrated design: strategies for architectural practice*. Hoboken, New Jersey: John Wiley & Sons, 2011.
24. Messinger PR, Stroulia E, Lyons K, et al. Virtual worlds—past, present, and future: New directions in social computing. *Decision Support Systems* 2009; 47(3): 204–228.
25. Whyte J. *Virtual reality and the built environment*. Abingdon, Oxfordshire: Routledge, 2007.
26. Vite C, Horvath A-S, Neff G, et al. Bringing Human-Centredness to Technologies for Buildings: An agenda for linking new types of data to the challenge of sustainability. In A. De Angeli, L. Chittaro, R. Gennari, et al. (eds), Proceedings of the 14th Biannual Conference of the Italian SIGCHI Chapter (pp. 1–8), 2021. Association for Computing Machinery. https://doi.org/10.1145/3464385.3464711
27. Lou EC, The Quan N and Park K. Guest editorial: Disruptive and innovative technologies for sustainable development in the built environment. *International Journal of Building Pathology and Adaptation* 2022; 40(3): 301–304.
28. Al-Qawasmi J and Hadjri K. Architecture in the digital age: The effect of digital media on the design, production and evaluation of the built environment. *Open House International* 2007; 32(2): 4–6.
29. Gross M and Do E. Integrating digital media in design studio: Six paradigms. Minneapolis: Proceedings of the American college schools of architecture conference, 1999.
30. Marble S (ed). *Digital Workflows in Architecture: Design–Assembly–Industry.* Berlin, Germany: Walter de Gruyter, 2012.
31. Wortmann T and Tunçer B. Differentiating parametric design: Digital workflows in contemporary architecture and construction. *Design Studies* 2017; 52: 173–197.
32. Hansmeyer M., 2017. Digital Grotesque II at Centre Pompidous 'Imprimer Le Monde'. https://dbt.arch.ethz.ch/project/digital-grotesque-at-centre-pompidou/(2017, accessed 27 Oct. 2022).
33. Leach N. In the mirror of AI: what is creativity? *Archit Intell* 2022; 1(1): 8–15.
34. Del Campo M, Carlson A and Manninger S. Towards hallucinating machines-designing with computational vision. *International Journal of Architectural Computing* 2021; 19(1): 88–103.
35. Steinfeld K. Significant others: Machine learning as actor, material, and provocateur in art and design. *The Routledge Companion to Artificial Intelligence in Architecture*. London: Routledge, 2021, pp. 3–12.
36. Tamke M, Nicholas P and Zwierzycki M. Machine learning for architectural design: Practices and infrastructure. *International Journal of Architectural Computing* 2018; 16(2): .123–143.
37. Belém C, Santos L and Leitão A. *On the impact of machine learning: architecture without architects?* Daejeon, Korea: 18th International Conference, CAAD Futures 2019, Proceedings, 2019.
38. As I, Pal S and Basu P. Artificial intelligence in architecture: Generating conceptual design via deep learning. *International Journal of Architectural Computing* 2018; 16(4): 306–327.
39. Yang Z, Jiang H and Zou L. 3D Conceptual Design Using Deep Learning. In Science and Information Conference (pp. 16–26). Cham: Springer, 2019.
40. Del Campo M. 3D Graph convolutional neural networks in architecture design. *Proceedings of the ACADIA Conference*. Online + Global: ACADIA, 2020.
41. Palamas G, Guerra AM and Møsbæk LD. AI_deation: A Creative Knowledge Mining Method for Design Exploration. *EAI Endorsed Trans Creat Tech* 2022; 9(3): e5.

42. Huang W and Zheng H. *Architectural drawings recognition and generation through machine learning*. Mexico City, Mexico: In Proceedings of the 38th Annual Conference of the Association for Computer Aided Design in Architecture, 2018, pp. 18–20.
43. Algeciras-Rodriguez J. *Stochastic Hybrids-From references to design options through Self-Organizing Maps methodology*. Lodz, Poland: Proceedings of the 36th eCAADe Conference, 2018, 1.
44. Miljković D. Brief review of self-organizing maps. *2017 40th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. New York: IEEE, 2017, pp. 1061–1066.
45. Meng LL, Graham J and Haeusler MH. *t-SNE: a dimensionality reduction tool for design data visualisation*. Bangkok, Thailand: Proceedings of the 25th CAADRIA Conference, 2020, pp. 629–638.
46. Lunterova A, Spetko O and Palamas G. Explorative visualization of food data to raise awareness of nutritional value. In *International Conference on Human-Computer Interaction* (180–191). Cham: Springer, 2019.
47. Harding J. *Dimensionality reduction for parametric design exploration*. Zurich, Switzerland: Conference: Advances in Architectural Geometry, 2016.
48. Lunterova A, Spetko O and Palamas G. Computational Aesthetics of the Collective Affective Dynamics of IMDB Movie Reviews. Porto, Lisbon: 10th Conference on Computation, Communication, Aesthetics & X: 10th Conference on Computation, Communication, 2022, p. 26.
49. Goodfellow I.. Nips 2016 tutorial: Generative adversarial networks. Online, *arXiv preprint,* https://doi.org/10.48550/arXiv.1701.00160, 2016
50. Van Den Oord A, Kalchbrenner N and Kavukcuoglu K. Pixel recurrent neural networks. In International conference on machine learning. https://proceedings.mlr.press/: *PMLR* 2018; 1747–1756.
51. Goodfellow I, Pouget-Abadie J, Mirza M, et al. Generative adversarial networks. *Commun ACM* 2020; 63(11): 139–144.
52. Creswell A, White T, Dumoulin V, et al. Generative adversarial networks: An overview. *IEEE Signal Process Mag* 2018; 35(1): 53–65.
53. Li C.L., Zaheer M., Zhang Y., Poczos B. and Salakhutdinov R. Point cloud gan. *arXiv preprint arXiv:1810.05795*, 2018.
54. Sanghi A, Chu H, Lambourne JG, et al.*Clip-forge: Towards zero-shot text-to-shape generation*. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. IEEE. 2022. pp. 18603–18613.
55. Zhang J, Chen X, Cai Z, et al.Unsupervised 3d shape completion through GAN inversion. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021. pp. 1768–1777.
56. Li R, Li X, Fu CWet al. Pu-gan: a point cloud upsampling adversarial network. Long Beach, California: icant others: Machine learning as actor, material, and provocateur in art and design. The Routledge Companion to Artificial Intelligence in Architec, pp. 7203–7212.
57. Achlioptas P, Diamanti O, Mitliagkas Iet al. Learning representations and generative models for 3D point clouds. *International conference on machine learning*. https://proceedings.mlr.press/: PMLR, 2018, pp. 40–49.
58. Valsesia D, Fracastoro G and Magli E. Learning localized generative models for 3d point clouds via graph convolution. In International conference on learning representations, 2018.
59. Shu DW, Park SW and Kwon J. 3d point cloud generative adversarial network based on tree-structured graph convolutions. In Proceedings of the IEEE/CVF international conference on computer vision, 2019, pp. 3859–3868
60. Yang X, Wu Y, Zhang K, et al. CPCGAN: A controllable 3D point cloud generative adversarial network with semantic label generating. *Proceedings of the AAAI Conference on Artificial Intelligence* 2021; 35(4): 3154–3162.
61. Qi CR, Su H, Mo K, et al. Pointnet: Deep learning on point sets for 3d classification and segmentation. *Proceedings of the IEEE conference on computer vision and pattern recognition*. New York: IEEE, 2017, pp. 652–660.
62. Yi L, Kim VG, Ceylan D, et al. A scalable active framework for region annotation in 3d shape collections. *ACM Trans Graph* 2016; 35(6): 1–12.
63. Hana XF, Jin JS, Xie J, et al. comprehensive review of 3D point cloud descriptors. *arXiv preprint arXiv:*1802.02297, 2, p.3, 2018

64. Horvath A-S. Assessing Site-Geometry for Architectural Design Using Graph Theory. In C Chiorean. (Ed.), *Proceedings of the Second International Conference for PhD students in Civil Engineering and Architecture: Building the Community of Young Researchers*. U.T. Press, pp. 611–619, 2014. https://doi.org/10.6084/m9.figshare.14555541.v1
65. Wandinger U. Introduction to Lidar. Springer, New York, NY, 2005.
66. Geiger A, Lenz P and Urtasun R. Are we ready for autonomous driving? the kitti vision benchmark suite. *2012 IEEE conference on computer vision and pattern recognition*. New York: IEEE, 2012, pp. 3354–3361.
67. Wu Z, Song S, Khosla A, et al.3D shapenets: A deep representation for volumetric shapes. In Proceedings of the IEEE conference on computer vision and pattern recognition, 2015.
68. Malinverni ES, Pierdicca R, Paolanti Met al. Deep learning for semantic segmentation of 3D point cloud. *International archives of the photogrammetry*. Germany: Remote Sensing & Spatial Information Sciences, 2019.
69. Croce V, Caroti G and De Luca L, et al. 2021. From the semantic point cloud to heritage-building information modeling: A semiautomatic approach exploiting machine learning. *Remote Sensing* 2021; 13(3): 461.
70. McNeel. https://www.rhino3d.com/, 2019.
71. Vestartas. Cockroach https://www.food4rhino.com/en/app/cockroach, 2020.
72. Bridson R. Fast Poisson disk sampling in arbitrary dimensions. *SIGGRAPH Sketches* 2007; 10(1): 1.
73. Chevi R. Growing Neural Gas Implementation. https://github.com/rendchevi/growing-neural-gas (2020, accessed 26 October 2022).
74. Holmström J. Growing neural gas: experiments with GNG, GNG with utility and supervised GNG. Master Thesis. *Uppasala*: Uppsala University, 2002. https://www.booru.net/download/MasterThesisProj.pdf. [Accessed 6 Nov. 2022].
75. Daszykowski M, Walczak B and Massart DL. On the optimal partitioning of data with K-means, growing K-means, neural gas, and growing neural gas. *J Chem Inf Comput Sci* 2002; 42(6): 1378–1389.
76. Yang X. DLNest. https://github.com/SymenYang/DLNest (2022, accessed 26 Oct. 2022).
77. Roudsari M.S. and Pak M., 2013. Ladybug: a parametric environmental plugin for grasshopper to help designers create an environmentally-conscious design. Proceedings of the 13th International IBPSA conference. Lyon. 2013. pp. 3128-3135
78. Kharbanda K, Papadopoulou I, Pouliou Pet al. *LearnCarbon-A tool for machine learning prediction of global warming potential from abstract designs*. Ghent, Belgium: Conference: eCAADe 2022: Co-creating the Future - Inclusion in and through Design, 2022, p. 601.
79. Schön DA. *The Reflective Practitioner: how professionals think in action*. Abingdon, Oxfordshire: Routledge, 2017.
80. Oxman R. Theory and design in the first digital age. *Design Studies* 2006; 27(3): 229–265.
81. Dunne A and Raby F. *Speculative everything design, fiction, and social dreaming*. Cambridge, Massachusetts: MIT Press, 2013.
82. Ozkaramanli D and Desmet P. (2016). Provocative design for unprovocative designers: Strategies for triggering personal dilemmas. In P. Lloyd and E. Bohemia (eds), *Proceedings of DRS 2016, Design + Research + Society - Future-Focused ThinkingDesign : Research Society 50th Anniversary Conference* (Vol. 1, pp. 2001–2016). London: The Design Research Society.
83. Marenko B. FutureCrafting. A Speculative Method for an Imaginative AI. In *AAAI Spring Symposium Series*. Palo Alto, USA: Association for the Advancement of Artificial Intelligence. pp. 419–422, 2018
84. Boden MA. What is creativity? *Creativity in human evolution and prehistory*. Abingdon, Oxfordshire: Routledge, 2005, pp. 27–55.
85. Boden MA. Computer Models of Creativity. *AI Mag* 2009; 30(3): 23–34.
86. Pouliou P, Horvath AS and Palamas G. Annotated point clouds of buildings: a segmented dataset of single-family houses. *Mendeley Data* 2022, doi:10.17632/3thtp7mc6z.2.
87. Pouliou P., Horvath A. S. and Palamas G.. Speculative Hybrids. *GitHub* 2023, doi:https://github.com/PanayotaPouliou/SpeculativeHybrids.