Representations in design computing through 3-D deep generative models

Başak Çakmak[1] [iD] and Cihan Öngün[2] [iD]

[1]Department of Digital Game Design, Istanbul Bilgi University, Istanbul, Turkey and [2]Graduate School of Informatics, Information Systems, Middle East Technical University (METU), Ankara, Turkey

Abstract

This paper aims to explore alternative representations of the physical architecture using its real-world sensory data through artificial neural networks (ANNs). In the project developed for this research, a detailed 3-D point cloud model is produced by scanning a physical structure with LiDAR. Then, point cloud data and mesh models are divided into parts according to architectural references and part-whole relationships with various techniques to create datasets. A deep learning model is trained using these datasets, and new 3-D models produced by deep generative models are examined. These new 3-D models, which are embodied in different representations, such as point clouds, mesh models, and bounding boxes, are used as a design vocabulary, and combinatorial formations are generated from them.

Introduction

This research investigates the potential of 3-D deep generative models within the field of design computing, specifically targeting the challenge of transforming sensory data from the physical environment into actionable design intelligence. The advent of high-fidelity 3-D scanning technologies and artificial neural networks (ANNs) has unlocked new possibilities for capturing and interpreting complex architectural data. However, the effective utilization of this data in design computing remains a significant challenge, marked by gaps in methodologies for data processing that can fully leverage the depth of information available. Central to our investigation is the ambition to address these methodological gaps by proposing a novel framework that integrates 3-D scanning data, such as LiDAR-generated point clouds, with deep learning algorithms to generate alternative design models. These models aim to enrich the design process with new forms and patterns derived from the real world, yet the translation of raw data into meaningful design elements requires innovative approaches that can navigate the intricacies of architectural references and the dynamics of part-whole relationships in design computing. The specific challenges this research seeks to overcome include the efficient segmentation of complex 3D point-cloud models into analyzable datasets, the application of encoder–decoder models for learning design patterns from these datasets, and the generation of new design alternatives through generative adversarial networks (GANs). These challenges emphasize the need for an integrated approach that can bridge sensory data capture and generative design. By situating our work within the broader context of design computing, this paper contributes to the ongoing discourse on the integration of machine learning (ML) techniques in the design process. It underscores the transformative potential of combining advanced computational methods with traditional design principles to foster a new era of design exploration and innovation. Through detailed experimentation and analysis, our research not only demonstrates the technical viability of our proposed approach but also reflects on its implications for design theory and practice, aiming to inspire further advancements in the field of design computing.

Background

The present research aims to use physical world structures to be decoded by ANNs to explore representations of design alternatives. The methodology used in this research is based on deep generative models. Therefore, in this section, 3-D deep generative models and related works in the field of design computing are discussed in order to distinguish the specific contribution of this research to the field.

Generative models and their evolution

Generative models aim to generate new data from the same distribution of given training data. With the advancements in ANN, deep generative models are proposed, which can outperform the previous studies (Goodfellow et al., 2014). GAN (Goodfellow et al., 2014), which is one of the state-of-the-art generative models, consists of 2 different ANNs that play an adversarial minimax

game against each other; generator G and discriminator D. The Generator aims to generate novel realistic samples, and the discriminator tries to distinguish between real and fake samples. At each iteration, the discriminator gets better at identifying fake samples, and the generator, using the feedback from the discriminator, generates more realistic samples to fool the discriminator.

Recent advancements in 3-D generative modeling

In the realm of existing research, Shi et al. (2023) delve into various facets of 3D generative models, offering an extensive overview of recent advancements and applications. This survey is pivotal in understanding the broader context of 3D generative model development and its potential impact in various fields, including design computing. Their insights provide a crucial foundation for the current study, underlining the evolving nature of generative modeling and its increasing relevance in the design domain. Zhang and Blasetti (2020) used 3-D models for training GAN. The study converts 3-D modes into 2-D images to use in training GANs as 2D images are more manageable for processing by GANs. Various popular generative models, such as Pix2Pix (Isola et al., 2017) and CycleGAN (Zhu et al., 2017), were used in the research to generate design options. In this alternative application of neural style transfer, the aim was to create a 3-D form by positioning 2-D outputs in different directions. Zhang and Blasetti (2020) aimed to generate 3-D models with StyleGAN (Karras et al., 2019), which is a state-of-the-art generative model that was trained with the reconstruction of 3-D models in various complex concepts. It was mentioned that the 3-D model is composed of changing images, and these images were used as a dataset to create new 3-D models on various complexity levels of input.

Applications in architectural design

In one of the researches on deep generative models, using a dataset that included house plans that have a specific architectural character, GAN was used to generate new plans that had the features of this architecture (Newton, 2019). Further, in the architectural context, Liu et al. (2019) produced 3-D model parts that have a reference to specific architectural styles, further combined to create more complex models that have various design references. After that, rendered images were used as a database for GAN. The image results were presented in the environments according to their concepts. In another study, Peng et al. (2017) worked on a dataset, including parts of 3-D models, to decode architectural space using ML and computer Vision. These models were the buildings of renowned architects. It was expected to recognize specific local compositions in these partial models to produce new compositions. Images of model parts were used for training a Neural Network, and new 3-D configurations were created by postprocessing on 2-D output.

Utilizing point clouds in 3-D generative strategies

Point clouds are a set of unstructured points in a 3-D coordinate system that represents real-world 3-D objects. It is mostly used in robotic applications and 3-D scanners like LiDAR. It is the most used capturing technique to digitalize real-world structures. To calculate the similarity of point clouds for reconstruction, the Chamfer Distance is used (Fan et al., 2017). It is the nearest neighbor distance metric for point sets. It is permutation invariant to work on unordered sets. Bidgoli and Veloso (2018) used an AutoEncoder for point clouds to provide a new generation of 3-D objects. They mentioned the advantages of using point cloud for 3-D representations in ML because it allows the production of samples both by using a digital model and by scanning physical objects using LiDAR Scanners.

Advancements in the current research

In this research, a GAN is used to generate new samples from the learned architectural character encoded in a physical structure. Also, an AutoEncoder is used in this study to learn and encode the common style of input samples. Since we aim to learn from the information of a real-world structure, we choose to work on point clouds instead of other 3-D representation methods such as Polygonal Meshes or 3-D Voxels. In most of the existing works, 2-D generative models were used due to the difficulties encountered in 3-D processing in ML. The present research examines the built-in architecture in detail to explore its alternative representations. The study uses generative models in a 3-D environment during the process of automation to decode real-world data and investigates the morphology that creates a character by conscious design decisions.

Research objectives

This research introduces an approach in generative design through the application of 3-D Deep Generative Models, significantly expanding the potential of ML in representing the unique morphology created by the amalgamation of design elements. The study is centered around the application and analysis of 3-D point cloud data, obtained through LiDAR Scanning, to accurately capture and transform the architectural structure into comprehensive datasets. These datasets are intricately manipulated to effectively represent the design features of real-world structures, focusing on the complex relationships and references that are integral to the architectural design.

A significant aim of this study is to identify and apply the appropriate information processing models, especially ANNs, for interpreting the complex data encapsulated in the physical structure of the design. These models are meticulously trained to decode and represent the nuanced aspects of the design contained within the datasets. By focusing on the parts of architectural structures, this research enhances the generation of a specific design vocabulary, addressing the limitations of current research that mainly relies on 2-D and abstract data. Our methodology stands out by harnessing real-world data, enabling the processing of 3-D model parts with ML to achieve spatial generations. Another key objective is to employ Deep Generative Models to foster the generation of alternative representations of the architectural design. This involves utilizing the trained models to generate diverse and meaningful samples that offer interpretations and insights for future generative tasks in design computing. Through this research, the aim is to contribute to the field of design computing by enhancing the understanding of how ML models can be applied in architectural contexts. This study serves as a stepping stone for future explorations into the use of ML in design computing and the development of advanced generative design systems.

Case study: application in architectural design

The approach underlying this descriptive research is experienced in a project. This project creates a platform for the study to be tested

and examined qualitatively in a chosen context by the use of 3-D Deep Generative Models and the use of datasets in various forms as mesh models and point cloud models. The data collection for the project is conducted from primary sources of the chosen context, The Faculty of Architecture at Middle East Technical University (METU). Original drawings of the chosen context are provided by the METU Directorate of Construction and Technical Services. The LiDAR Scanner model was developed by the Photogrammetry Laboratory of the Faculty of Architecture at METU, and mesh models are our own production. The stages of the project are described in the workflow diagram in Figure 1.

Conceptual framework

This research experiments to process design context with ML using point cloud data directly obtained from the physical environment. In order to analyze the design character in the 3-D contexts and to investigate its use for alternative representations through ML, a 3-D model is deconstructed and processed. Then the new 3-D models produced by the ML model are examined. Different combinations of design elements are selected, and spatial parts are analyzed through 3-D deep learning algorithms. For this purpose, the ML model is trained by using 3-D parts of the design system to analyze
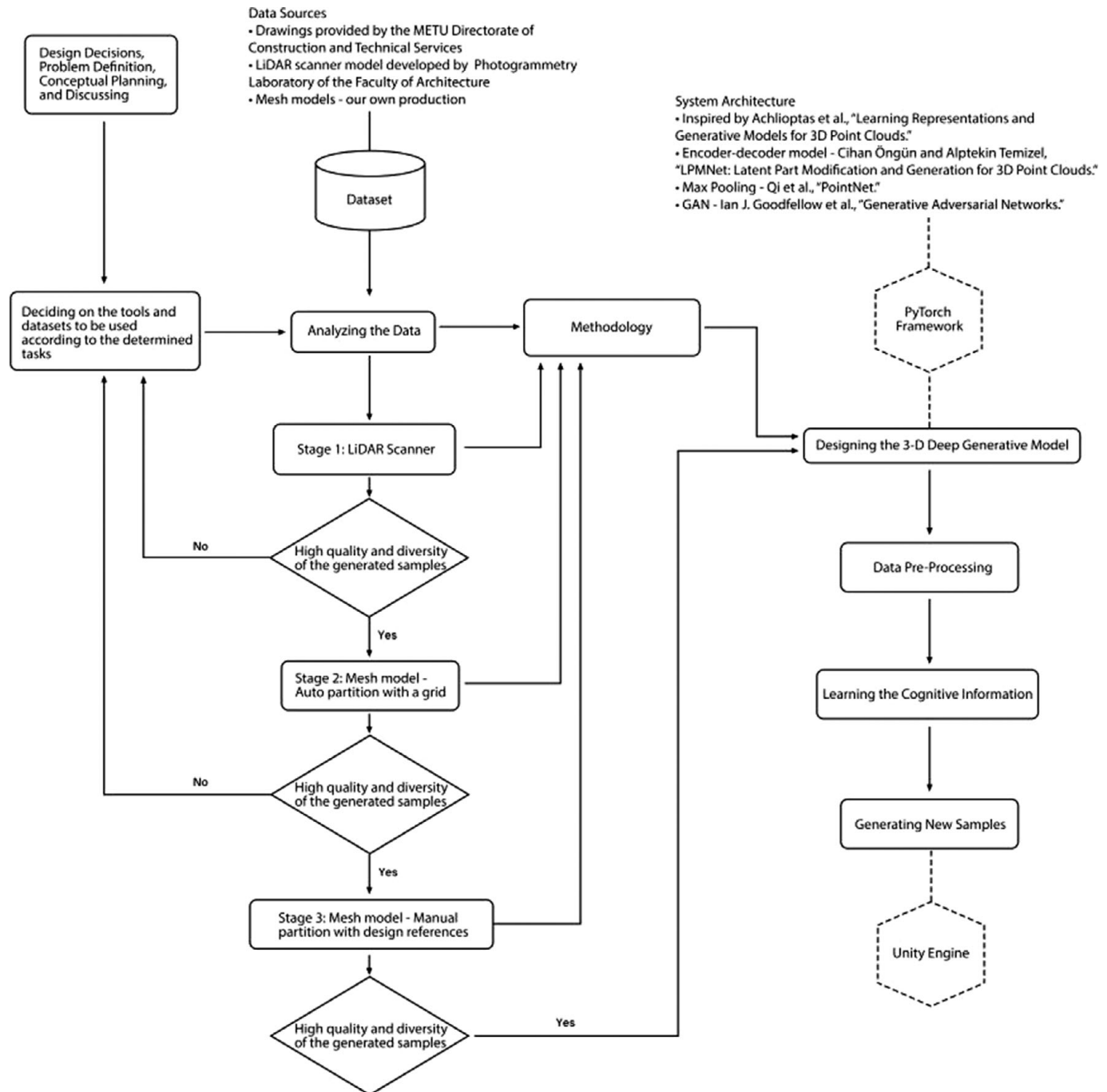

Figure 1. Workflow diagram.

the information about the features of the whole system encoded in the parts. Later, a generative model is used to produce new samples reflecting a design representation.

Contextual analysis

The specific architectural form and level of complexity chosen for this study are critical in demonstrating the capabilities of 3-D deep generative models. The METU, selected as the primary case, exemplifies a structure with meaningful and repetitive combinations of design elements, exhibiting a strong and distinctive architectural character. This particular case was chosen not only for its architectural significance but also for its diverse array of design elements that represent a challenging and insightful subject for analysis.

The METU Faculty of Architecture building presents a unique blend of architectural forms, making it an ideal subject for exploring the nuances of spatial design and generative modeling. The building's design composition is characterized by a robust articulation of solids, providing a rich context for applying and testing our 3-D modeling techniques. The complexity and distinctiveness of the building's design elements allow for a thorough investigation into the potential of 3-D generative methods in capturing and reinterpreting architectural characteristics.

Moreover, the choice of this building was also influenced by the availability of comprehensive data from primary sources. The accessibility of detailed architectural data, including the views from the 3-D mesh model of the building as shown in Figure 2, has been instrumental in the development and validation of our methodology. By utilizing a well-documented and architecturally significant structure, the study ensures a robust and relevant application of its generative modeling techniques, aiming to contribute substantial insights into the field of design and computation.

Data collection and analysis

This project explores the transformation of design data for use in ANNs, focusing on architectural references and volumetric relationships to reflect the design process. This investigation unfolds through consecutive stages in a comprehensive workflow, each building upon the previous to refine and enhance the dataset used for training our ML model.

Our approach meticulously segments the 3D model of a building into subunits, facilitating a focused analysis of design elements within these partitions. This approach allows for the extraction of detailed features from each subunit, contributing to a layered understanding of the architectural structure. Each subunit is analyzed independently, enabling the identification of unique architectural features and design principles at a granular level. This segmentation into subunits is central to our methodology, allowing us to explore the intricate details of the architectural design and its components. While this approach effectively captures the detailed features of individual subunits, it inherently presents a challenge in synthesizing these findings to extract global features that represent the entire building model comprehensively.

Our methodology primarily focuses on the in-depth analysis of the subunits, aiming to understand the building's architectural essence through its components. This focus on subunits allows for a deep, detail-oriented exploration of architectural elements, though it may limit the scope for capturing the building's global architectural features in their entirety.

Stage 1: 3D scanning with LiDAR

The selected design context was scanned with the LiDAR Scanner tool, which uses laser pulses to detect the distance of an object's surface. In this way, a highly detailed 3-D point cloud model was created for the chosen building. LiDAR provides the coordinate values of all the details of the building in real dimensions. This data was divided into smaller 3-D subunits that allowed detailed analysis of information (Figure 4). ML model was trained with the subunits of this 3-D point cloud model. Using real scans can help to decode the underlying system of the design process and provide experimental results on real-world data.

The LiDAR data has some problems regarding the uniformity and density of the points. The point density is higher near locations where the LiDAR device is located. This causes a non-uniform distribution of the points among the 3-D model. This also creates visible dense circles around the LiDAR device on the point cloud. ML systems tend to learn dense areas better than sparse areas. Thus, it may cause problems regarding reconstruction and generation quality. An extra pre-processing step (spatially uniform point sampling) is applied to make LiDAR data more uniform.
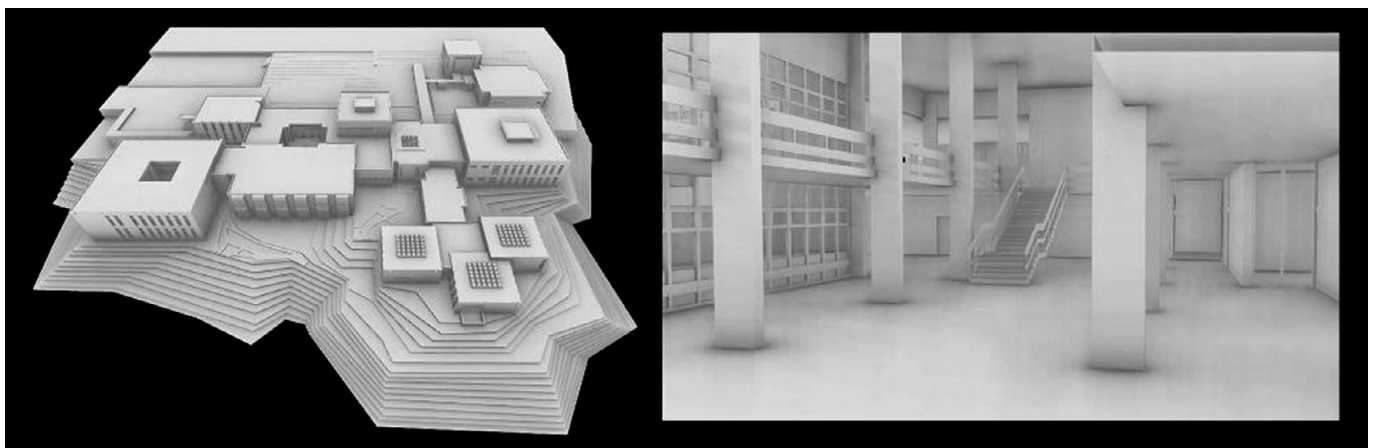

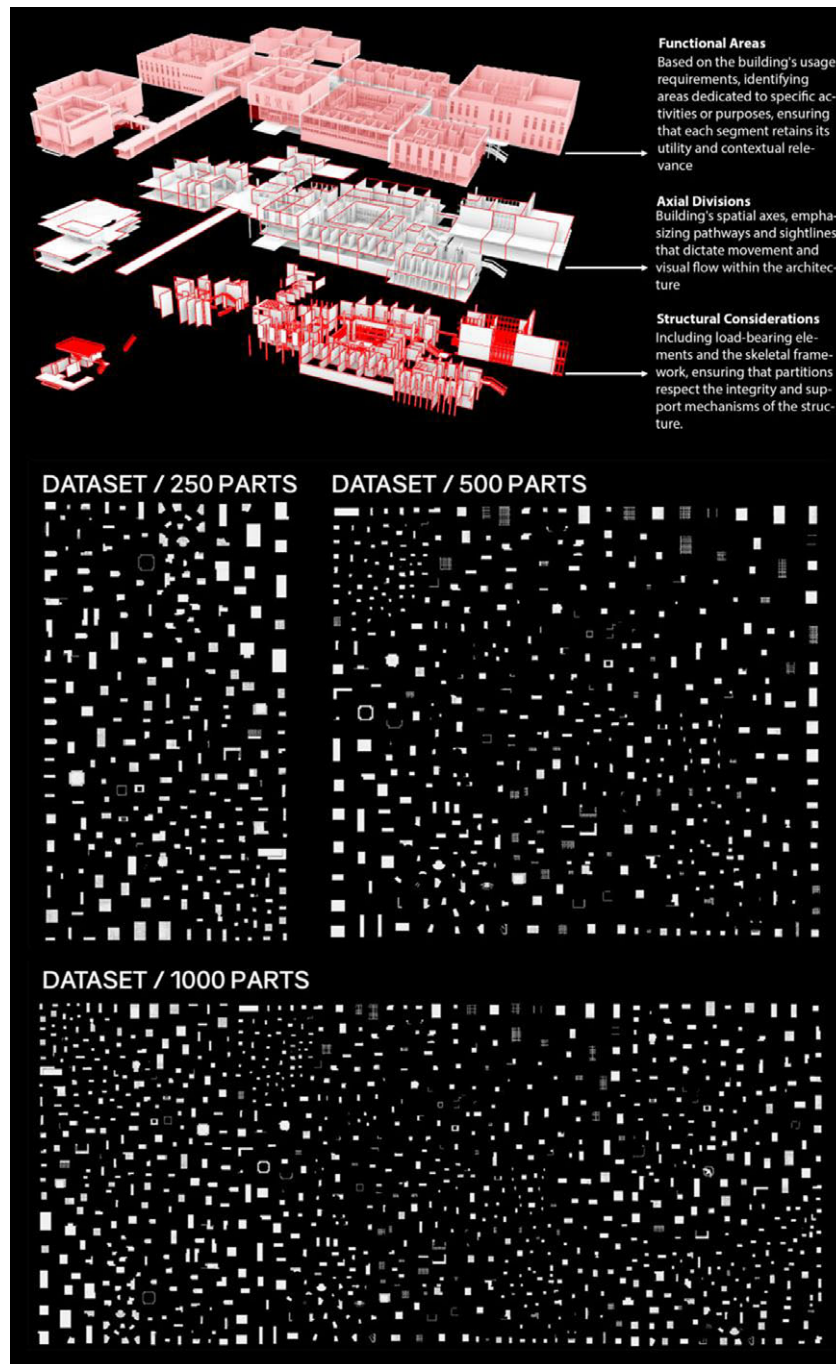**Figure 2.** A 3-D mesh model of the Faculty of Architecture at METU.

Functional Areas
Based on the building's usage requirements, identifying areas dedicated to specific activities or purposes, ensuring that each segment retains its utility and contextual relevance

Axial Divisions
Building's spatial axes, emphasizing pathways and sightlines that dictate movement and visual flow within the architecture

Structural Considerations
Including load-bearing elements and the skeletal framework, ensuring that partitions respect the integrity and support mechanisms of the structure.

DATASET / 250 PARTS   DATASET / 500 PARTS

DATASET / 1000 PARTS

Figure 3. Partitions according to the references of design elements.

Stage 2: 3D mesh partitioning with a grid

At this stage, the mesh model was partitioned with a 3-D grid. Figure 4 illustrates the division of the structure's point cloud and mesh model using a grid-based volume, along with the resulting dataset obtained after the grid partitioning process. This is a basic process for partitioning the buildings with respect to the given part count. These parts were used for training the ML model. When the results were examined, some generated models were very similar or as same as some parts of the dataset. When a dataset, which consists of elements that are similar to each other, is used for training a generative model, the model starts to memorize the elements in the dataset and repeat them instead of understanding the hierarchical relationships of these elements and making new productions (Achlioptas et al., 2018). Also, this automatic splitting causes unrelated and unrealistic parts, such as unconnected parts or elements from different spaces. The produced parts do not follow a meaningful pattern or a useful representation. The ML model imitates the dataset by generating unconnected or meaningless parts. This method reduces the quality and diversity of the generated samples.
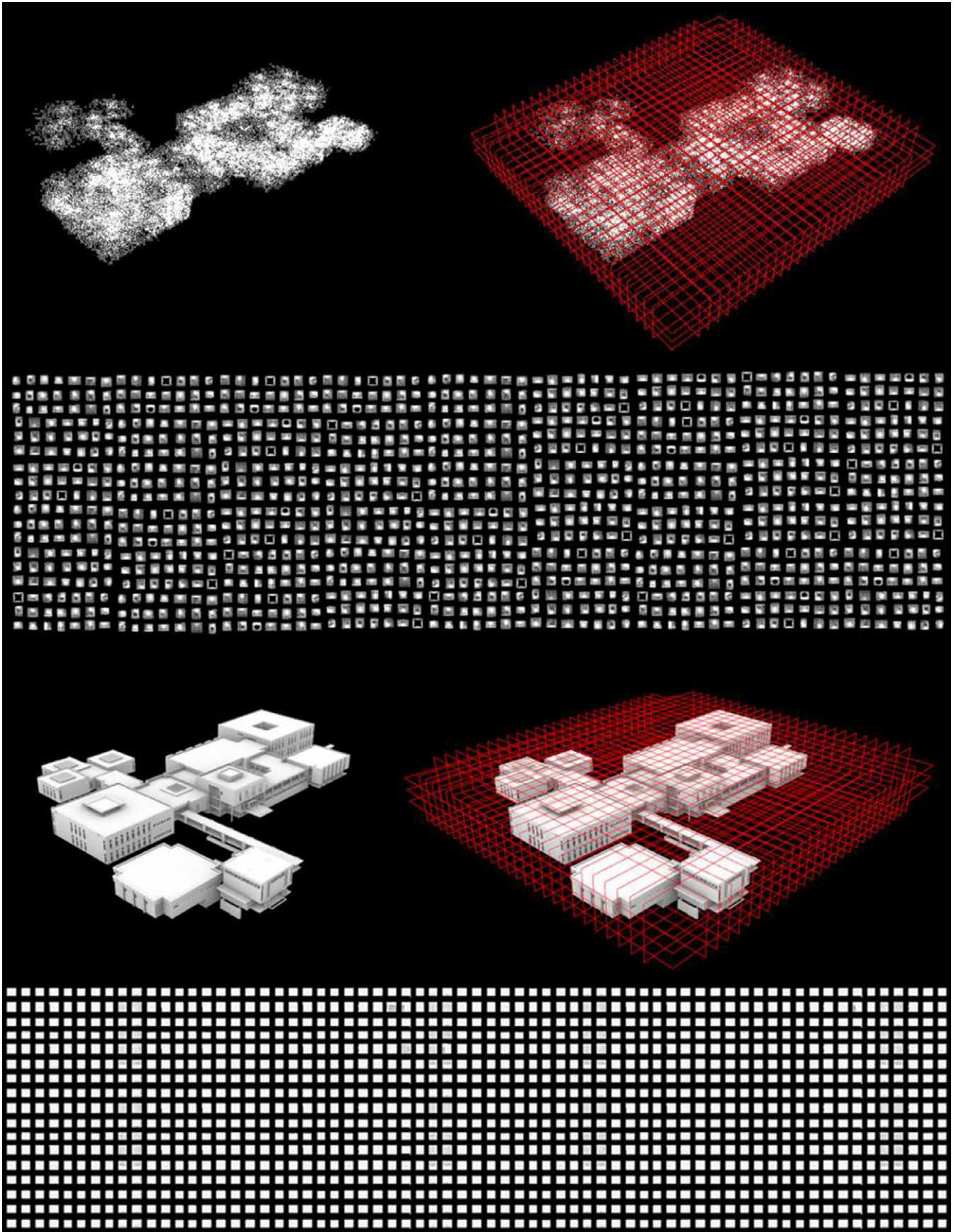
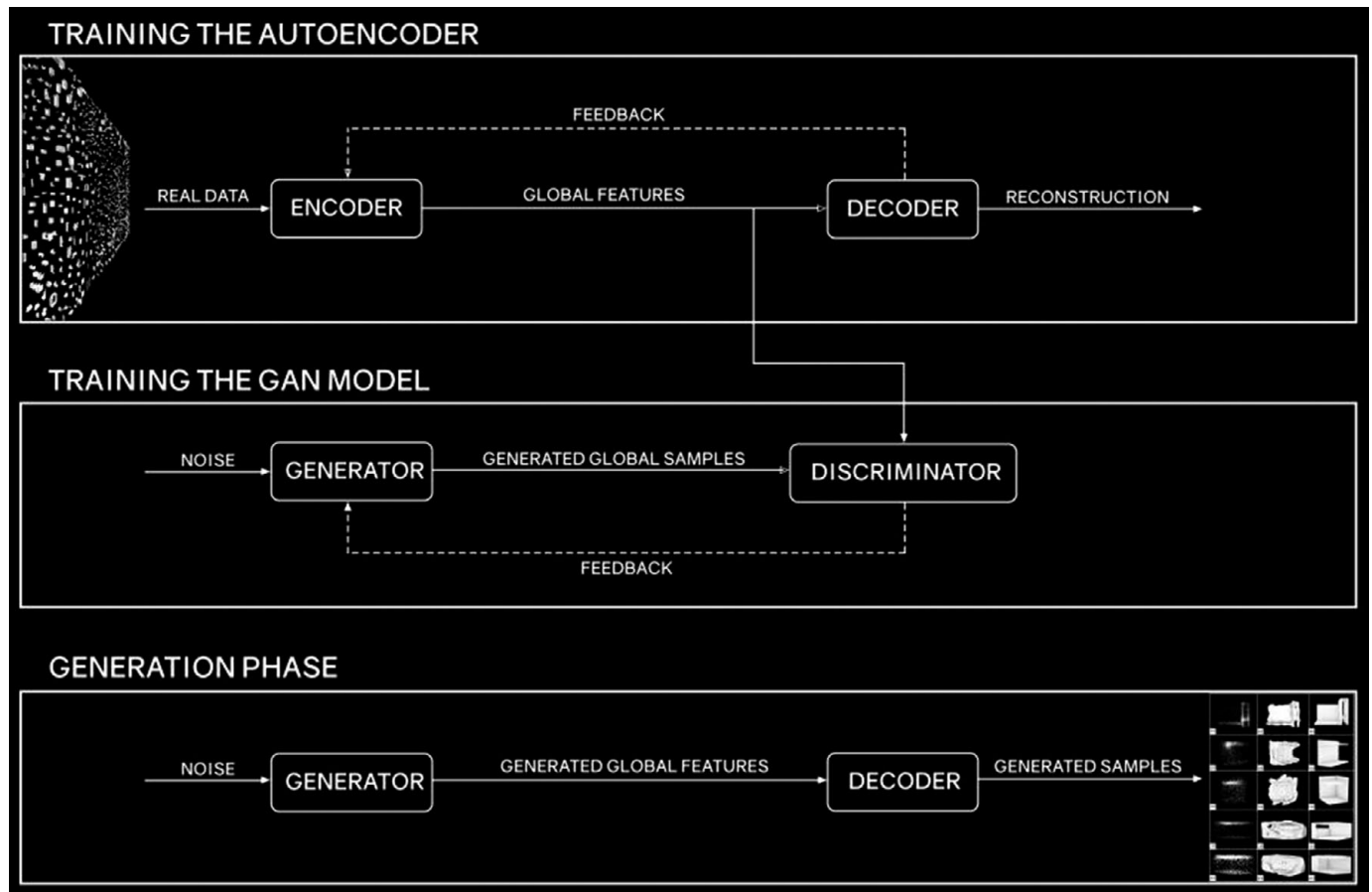**Figure 4.** Dividing the 3-D point cloud model and mesh model into 3-D subunits.

Figure 5. Methodology.

Stage 3: Design reference-based partitioning

After experimenting with partitioning using a grid, we create the dataset with logical partitions according to the references of design elements that carry the representation of the design process. In this context, design elements are interpreted through the lens of the grid system, which acts as a primary reference for the architectural configuration of spaces. These elements encompass the junction points of functionality and structure, where architectural spaces are crafted with consideration of their purpose and relationship to the overall grid. This method ensures that each partitioned segment is reflective of the building's functional layout and architectural intent. Moreover, the structural segmentation of the architecture plays a pivotal role in our partitioning process. It involves dissecting the building into meaningful components based on the structural integrity and load-bearing schema dictated by the grid. This approach allows for the identification of structural elements such as beams, columns, and load-bearing walls, which serve as critical references for segmenting the building into logical parts. With this approach, the whole building was divided into parts gradually, and the results were analyzed at each stage. First, the building was divided into 250 main parts that refer to the whole volume. Later, these 250 pieces were split into 500 pieces with references within themselves. Finally, all parts were brought together, and the parts were transformed into 1,000 subparts containing different relational combinations to allow the machine to focus more on the details. Datasets that are gradually divided into subspaces according to design references are displayed in Figure 3. The dataset consists of meaningful parts, each with a representation and uniform distribution of points and surfaces.

Methodological approach

The methodology of the study is inspired by Achlioptas et al. (2018). First, the data is pre-processed to feed the network, as explained in the Data preprocessing techniques Section. The data is fed to an encoder–decoder for learning the underlying structure of input samples. Then, a GAN is used to generate new samples from the learned structure. The encoder-decoder and the GAN models are explained in Learning design patterns: the encoder–decoder model section and Sample generation: employing GAN section, respectively. All implementations are done with the PyTorch framework on an Nvidia RTX 2070 GPU and are imported into the Unity Engine. The methodology of the study is demonstrated in Figure 5. The system architecture is visualized in Figure 6.

Data preprocessing techniques

The selected building is modeled in 3-D format. After partitioning the building model as explained in Data collection and analysis section, there are 250, 500, and 1,000 parts in 3-D mesh format for
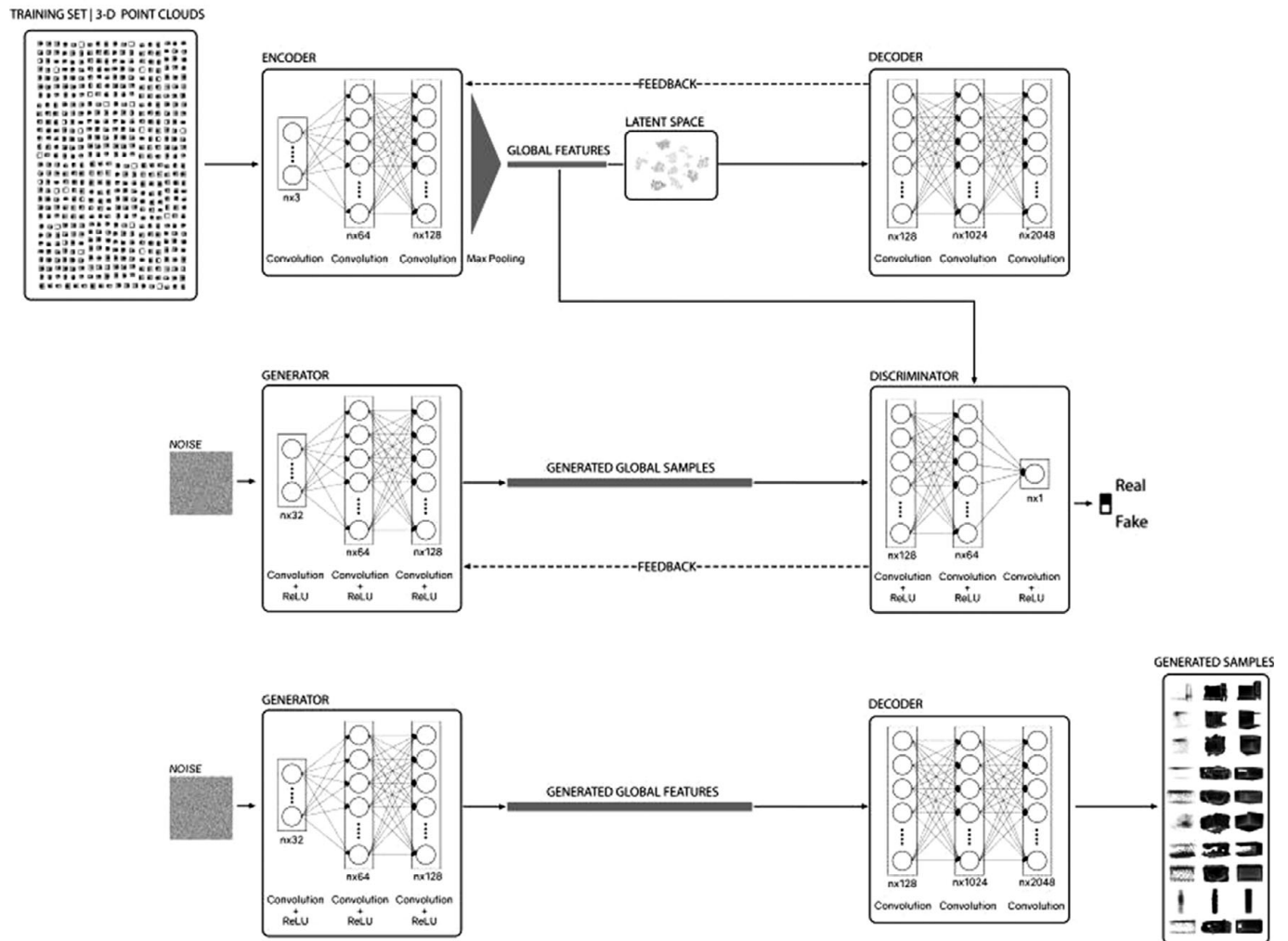
Figure 6. System architecture.

different experiments. The 3-D parts are converted to 3-D point clouds using uniform point sampling. Uniform point sampling is basically selecting a certain number of points to represent a surface. There must be at least 3 points to represent a triangle surface, and more points can be located using linear interpolation on the surface. Considering the mesh faces have different surface areas, to represent each surface uniformly, the resolution (point density) is set for each surface to its proportional area. All 3-D point cloud samples in the dataset are set to have 1,024, 2,048, 4,096, or 8,192 points for different experiments. After the conversion, all samples are positioned at the center of the coordinates and scaled into the unit cube. The dataset is randomly divided into train, validation, and test subsets with 80%, 10%, and 10% ratios, respectively.

Learning design patterns: the encoder-decoder model

Our first aim was to learn the underlying structure of the design system from the dataset. An encoder–decoder model (Achlioptas et al., 2018) (Figure 7) is employed to encode the real data and form a latent space that represents the design context. This latent space consists of learned features and similarities of the dataset. Since the dataset is formed from a single building, we expect the model to learn the similarities from the parts of the building.

The encoder model (Öngün and Temizel, 2021) is inspired by PointNet (Qi et al., 2017). It is a 3-layer 1-D convolutional network with feature sizes (3, 64, and 128). It extracts the features for each point having three dimensions for the x, y, and z-axis. Then a maxpooling is applied as explained in Qi et al. (2017) to extract the global feature (code) that represents the point cloud model. The input and feature transform subnetworks are omitted since the input data is already aligned and scaled. All extracted global features form a latent space that represents the underlying style and similarities of the dataset.

The global features are then decoded using a 3-layer Fully Connected Network (128, 1,024, and 2,048). The reconstruction loss is calculated with Chamfer Distance (Fan et al., 2017) between the real and reconstructed point clouds. The network is trained end-to-end using Adam (Kingma and Ba, 2015) optimizer with the reconstruction loss and a learning rate of 5×10^{-4} for 1,000 epochs. The reconstruction loss is around 10×10^{-4}, which indicates a good reconstruction performance with minimal error.

Sample generation: employing GAN

A GAN (Figure 8) is employed for generating new samples in the learned underlying structure of the design. The extracted global
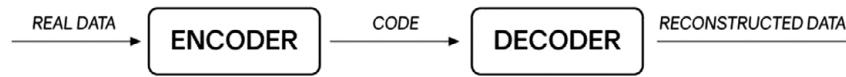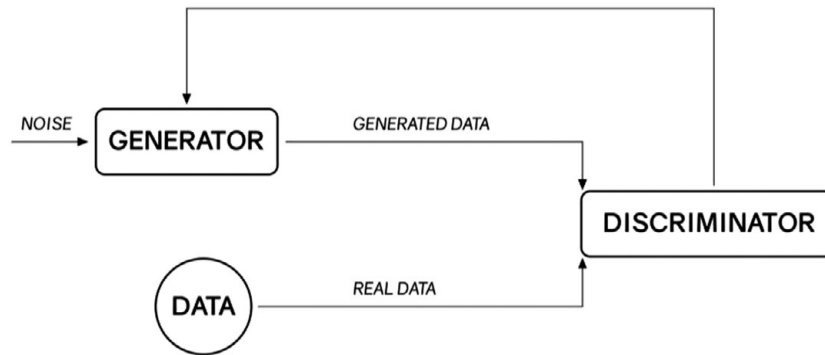
Figure 7. The AutoEncoder model.

Figure 8. The generative adversarial network (GAN) model

Table 1. The best WGAN results from the LPMNet (Öngün and Temizel, 2021) and our results

	Reconstruction loss Chamfer distance (×10⁻⁴)	Evaluation of the generated samples		
		MMD	%Cov	JSD(×10⁻²)
Ours	10.51	3.92	67.83	3.01
LPMNet	8.07	4.76	60.93	4.17

features are fed to the discriminator to train the discriminator alongside the generated global features using concatenation. The generator is trained by the feedback from the discriminator to make the generated samples more realistic. The generated global features are then decoded using the trained decoder to get 3-D point clouds that represent the real-world data.

Both the generator and the discriminator are 3-layer fully connected networks with (32, 64, and 128) and (128, 64, and 1) sizes, respectively. All layers have ReLU activation functions followed by batch normalization layers except the output layers. The Generator input is sampled from a Normal distribution. A WGAN (Arjovsky et al., 2017) objective function is used for training with better stability and diversity. Adam optimizer is used with learning rates of 5×10^{-4} and 1×10^{-4} for Generator and Discriminator, respectively. The code and more details about the models can be found in the baseline study (Öngün and Temizel, 2021).

Results and evaluation

In the study, the dataset is prepared and preprocessed to work with the proposed model. The AutoEncoder model is trained to evaluate the dataset and the learning ability of the model. The reconstruction loss between input and output is calculated using Chamfer Distance (Fan et al., 2017) which is a nearest neighbor distance metric for point sets. The reconstruction loss is 3.07×10^{-4} for training and 10.51×10^{-4} for testing, indicating a visually good reconstruction with a minimum loss between input and output.

The generated parts are evaluated using the same evaluation metrics of the LPMNet (Öngün and Temizel, 2021), which is used as a baseline for encoding–decoding the dataset and generating new parts. Our results are similar to the reconstruction loss of the LPMNet (8.07×10^{-4}), supporting our claim that our dataset serves well for the purpose of the encoding–decoding design system. The comparison with the baseline (Öngün and Temizel, 2021) shows that a better reconstruction can be achieved with a bigger and more diverse dataset by avoiding overfitting.

The results can be seen in Table 1. Coverage (Cov) measures the percentage of representation of generated parts in the input dataset. High coverage means that generated parts have high diversity to represent all different classes of data samples in the input dataset. Minimum matching distance (MMD) is the average of distances between the most similar samples in the input and generated sets. While low MMD means the samples are in the same scope and class, 0 means they are completely the same. Jensen- Shannon Divergence (JSD; derived from Kullback–Leibler divergence (Kullback and Leibler, 1951)) is a metric to calculate distances between probability distributions. In this study, it is used to measure if the generated samples occupy a similar scale, rotation, and location as the input set. The results are again comparable to the LPMNet, indicating that our case is suitable to generate new samples reflecting the features of the design character. The quantitative results support visual results that the generated parts are meaningful enough for forming alternative representations of the design system.

The 3-D Deep Generative Model trained with the point cloud data of building parts can produce new 3D models. The generated samples can be seen in Figure 9. The generated data is visualized in raw point cloud form first, as shown in the first columns of the tables in Figure 9. Then the generated point clouds are automatically transformed to mesh form with the Poisson Surface Reconstruction (Kazhdan et al., 2006) method to analyze the surfaces and the general connected structure of the data (see second columns in Figure 9). In our methodology, the mesh models represented in the second column of Figure 9 undergo a stage of manual post-processing, where they are flattened, as shown in the third column of Figure 9. In the progression from the generative output seen in
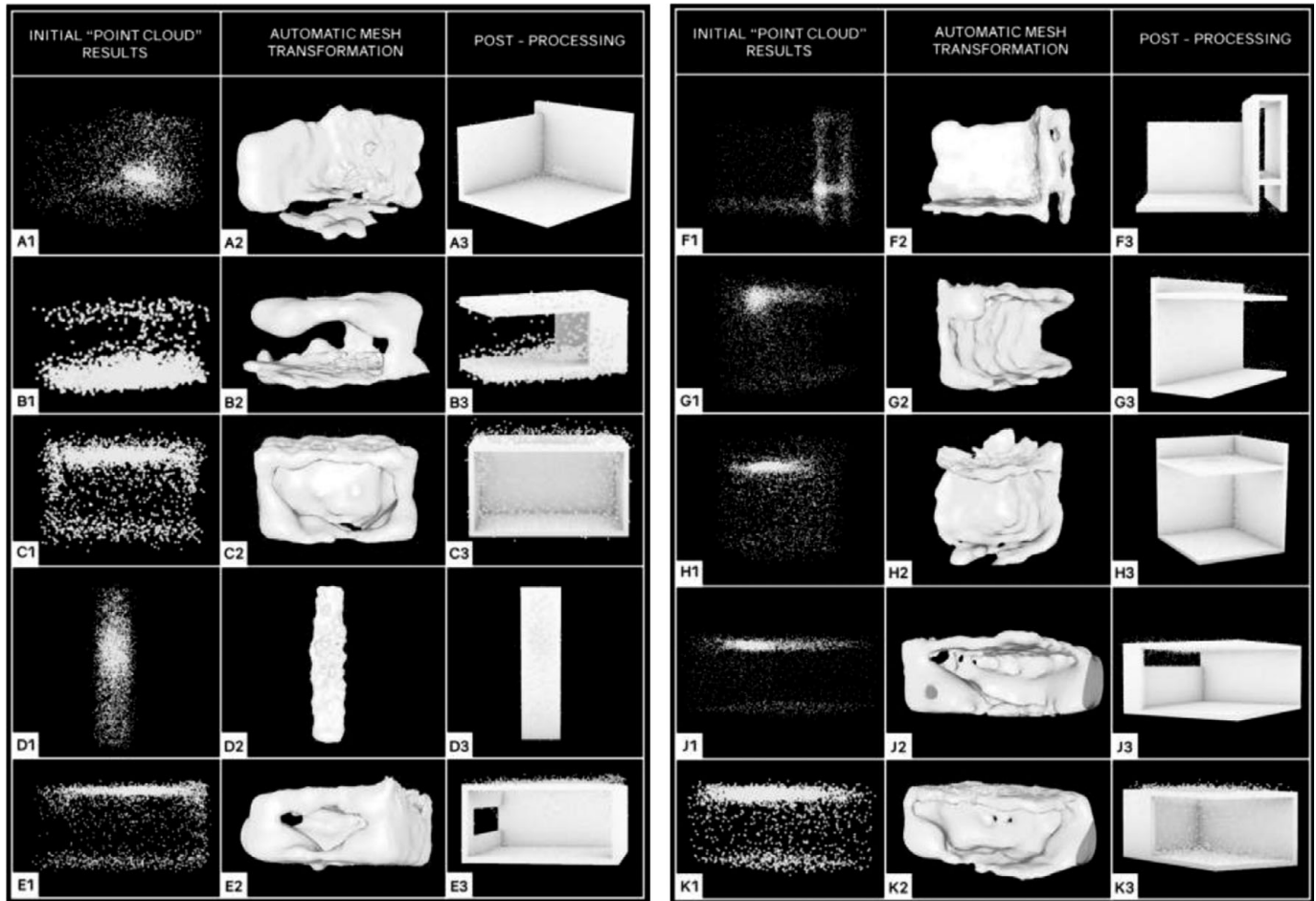
Figure 9. 3-D productions of the ML model in different abstraction levels.

column 2 of Figure 9 to the refined interpretations presented in column 3, a nuanced approach to post-processing was undertaken, meticulously guided by the initial forms generated by the GAN. These GAN-generated models served as the foundational blueprints upon which further transformations were applied, primarily aimed at enhancing architectural feasibility and aesthetic value. The focus was on achieving more straightforward and plain surfaces, a design attribute often sought in contemporary architectural practices for its visual clarity and structural efficiency. To accomplish this, we employed sculpting tools available in 3D modeling software. These tools allowed us to manipulate the mesh geometry directly, enabling precise control over the form and surface characteristics of the models.

The sculpting process involved selectively smoothing and flattening the surfaces, effectively transforming forms generated by the GAN into models with cleaner lines and more defined geometries. This manual intervention is essential for analyzing and employing various spatial representations at different levels of abstraction. It's important to recognize that this manual aspect of post-processing signifies an intriguing interplay between the generative capabilities of our models and the indispensable role of human expertise in the design process. Although the generative process effectively provides diverse and complex design outputs, the current state of technology necessitates human judgment to refine these outputs, aligning them with practical and aesthetic design considerations. This process is inherently subjective, reflecting the unique perspectives and

creativity of the designers. By manually adjusting and refining the generated models, designers can infuse their vision and expertise, shaping the raw computational output into forms that resonate with human sensibilities in architecture and design. The preservation of the original GAN-generated 3-D point cloud models in the postprocessed third columns of Figure 9, as well as in the sample scene in Figure 10, demonstrates our commitment to maintaining the integrity of the generative output while highlighting the transformative impact of manual post-processing. The spatial configurations of the generated samples, as exhibited in Figure 10, underscore their potential to foster a diverse range of architectural scenes and forms. This underscores the importance of the designer's role in the generative design process, suggesting that while the automation of generative models is advancing, the creative and subjective input of human designers remains a vital component in realizing the full potential of these technologies in design.

The results in Figure 9 are not merely theoretical representations but function as a practical design vocabulary. This vocabulary facilitates the exploration of new forms and spaces by combining different generated elements in a systematic manner. For instance, designers can use these models as blocks to conceptualize and visualize new layouts or modifications to existing formations. This approach allows for a more dynamic and innovative design process, where various permutations of the generated models can be assessed for aesthetic, functional, or structural suitability. In practice, this can lead to novel design solutions that are both inspired by traditional forms and
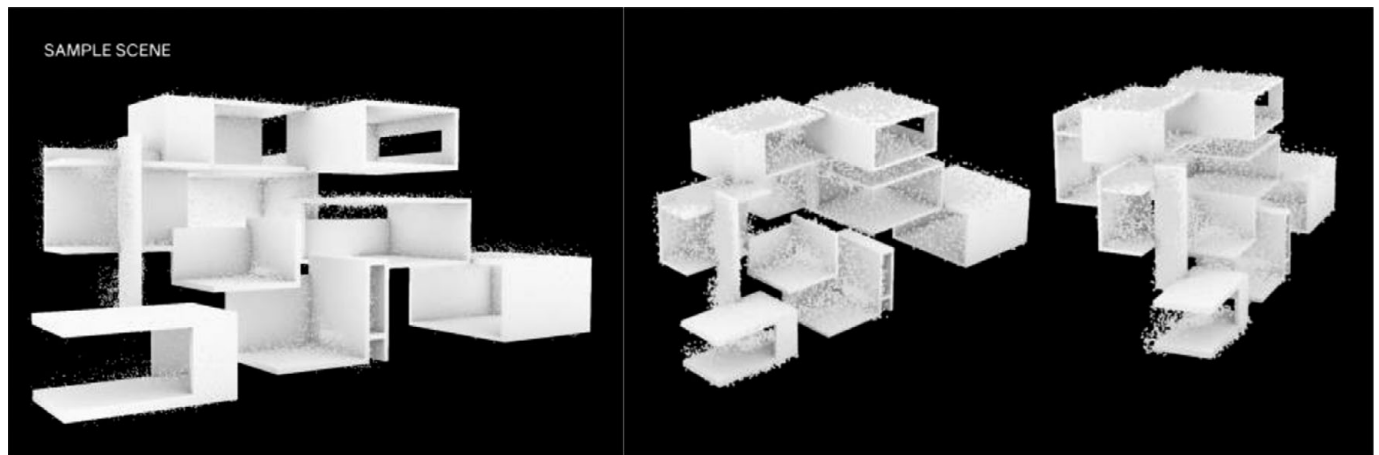
Figure 10. A Sample scene with spatial configurations of the productions.

adapted to contemporary needs. Furthermore, the combinatorial use of these generated models, as illustrated in Figure 10, offers a method for designers to interact with and modify the generated outcomes, injecting human creativity and expertise into the design process. This human-machine collaboration in design could be particularly beneficial in early-stage conceptualization, where rapid iteration and exploration of ideas are crucial.

Overall, the generative model developed in this study provides a versatile toolset for designers, enabling them to experiment with an array of design possibilities that were previously unattainable or time-consuming to explore. This method stands to significantly impact the design process, opening up new avenues for creativity and innovation.

Discussion

The design of 3-D deep generative models with the datasets of building parts demonstrates the practical application of this method in architectural contexts. The potential limitation of this approach lies in the balance between capturing exhaustive global features of the entire structure and delving deeply into the specific details provided by individual subunits. While our method enriches the design analysis by highlighting the distinct characteristics of its parts, it might constrain the holistic representation of global features as a unified entity. Future work could explore methodologies that integrate the detailed analysis of subunits with strategies for synthesizing these insights to more directly capture global features, offering a comprehensive understanding of both the parts and the whole of architectural structures.

In exploring the potential of 3-D deep generative models, this research directly engages with practical design scenarios by applying the generated models to the refurbishment and adaptive reuse of existing buildings. The practical design variables include spatial configurations, structural integrity, and historical preservation criteria, which are crucial in architectural redesign processes. By leveraging LiDAR-generated point clouds and ANNs, our approach facilitates the detailed analysis of physical structures, enabling designers to explore a multitude of design options within predefined constraints such as building codes and preservation guidelines.

This methodology not only enhances the design process but also supports the creation of dynamic, interactive environments, where historical accuracy and architecture are paramount. While the

initial findings suggest potential applications beyond architecture —such as in procedural content generation and engineering simulations—these extensions are speculative and require further empirical validation. The mention of these potential applications is intended to highlight the versatility of our approach and the breadth of its possible impact. As we set a foundation for future exploration, our research introduces a perspective on the use of ML for engaging with complex 3-D data in design. It opens discussions on spatial representation and the practical implications of integrating advanced computational models into the design process. This study, therefore, contributes to the ongoing dialogue in generative design and ML, offering insights while acknowledging the need for continued empirical research to fully realize and extend the methodologies and applications discussed.

Conclusions and future directions

This research provides an investigation for a generative design understanding by decoding and learning from the physical environment. The research hence contributes to the representation of design alternatives using real-world design data. In the stages of data collection and analysis, a 3-D mesh model and sensory data collected from a built-in structure with a LiDAR Scanner are used. The datasets are produced by transforming the design data in accordance with the part-whole relationship and reference system, reflecting the design process. These datasets, which represent the 3-D nature of the design features and provide the information of coordinates, are then fed to an AutoEncoder to learn the underlying structure. A deep generative model is used to generate new representations through the learned design character encoded in the physical structure. The generated samples can be analyzed in different forms, such as point clouds and 3-D meshes. The results show that the generated samples are meaningful for creating a design vocabulary in order to produce combinatorial formations for further generative tasks.

We aim to extend this study by experimenting with various cases to provide a better analysis of the proposed model. To further this research, the proposed model is planned to produce more results in the current case and be adapted to different design contexts. The versatility of our proposed method would find significant applications in the field of heritage building information modelling (HBIM) that greatly benefit from the accurate digital representation and analysis of architectural elements. In HBIM, our approach can

be instrumental in creating detailed libraries of architectural elements from various historical periods. These libraries can serve as foundational assets in preserving cultural heritage, enabling the precise modeling of historical buildings for maintenance, restoration, and educational purposes.

As we look to the future, the application of 3-D deep generative models in architectural design, HBIM, and game design presents a promising avenue for interdisciplinary research. The integration of these models with HBIM offers a novel approach to preserving and interacting with historical architecture, providing a bridge between traditional architectural practices and contemporary digital technology. Furthermore, the exploration of these models in game design opens up new possibilities for creating immersive, historically accurate virtual environments, pushing the boundaries of what is currently achievable in digital content creation.

References

Achlioptas P, **Diamanti O**, **Mitliagkas I and Guibas L** (2018) Learning representations and generative models for 3D point clouds. In *Proceedings of the 35th International Conference on Machine Learning (PMLR)*. Stockholm, Sweden: PMLR.

Arjovsky M, **Chintala S, and Bottou L** (2017) Wasserstein generative adversarial networks. In *Proceedings of the 34th International Conference on Machine Learning (PMLR)*, Sydney, Australia: PMLR.

Bidgoli A and Veloso P (2018) Deepcloud. The application of a data-driven, generative model in design. In *Proceedings of the 38th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA 2018)*, Mexico City, Mexico: ACADIA.

Çakmak B (2022). *Extending design cognition with computer vision and generative deep learning.* MArch Thesis, Ankara, Turkey: Middle East Technical University.

Fan H, **Su H and Guibas L** (2017) A point set generation network for 3D object reconstruction from a single image. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Honolulu, HI, USA: IEEE.

Goodfellow IJ, **Pouget-Abadie J**, **Mirza M**, **Xu B**, **Warde-Farley D**, **Ozair S**, **Courville A and Bengio Y** (2014) Generative adversarial networks. Preprint, arXiv:1406.2661v1.

Isola P, **Zhu JY**, **Zhou T and Efros AA** (2017) Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA: IEEE.

Karras T, **Laine S and Aila T** (2019) A style-based generator architecture for generative adversarial networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, USA: IEEE.

Kazhdan M, **Bolitho M, and Hoppe H** (2006) Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing (SGP 2006)*, Vienna, Austria: Eurographics Association.

Kingma DP and Ba J (2015) Adam: A method for stochastic optimization. In *3rd International Conference for Learning Representations (ICLR)*. San Diego, CA, USA: ICLR.

Kullback S and Leibler RA (1951) On information and sufficiency. *The Annals of Mathematical Statistics* **22**, 79–86.

Liu H, **Liao L and Srivastava A** (2019) An anonymous composition design optimization through machine learning algorithm. Ubiquity and autonomy. In *Proceedings of the 39th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA 2019)*, Austin, TX, USA: ACADIA.

Newton D (2019) Deep generative learning for the generation and analysis of architectural plans with small datasets. In *Proceedings of 37 eCAADe and XXIII SIGraDi Joint Conference*, São Paulo: Blucher.

Öngün C and Temizel A (2021) LPMNet: Latent part modification and generation for 3D point clouds. *Computers & Graphics* **96**, 1–13.

Peng W, **Zhang F and Nagakura T** (2017) Machines' perception of space. Disciplines and disruption. In *Proceedings Catalog of the 37th Annual Conference of the Association for Computer Aided Design in Architecture (ACADIA 2017)*, Cambridge, MA, USA: ACADIA.

Qi CR, **Su H**, **Mo K and Guibas LJ** (2017) PointNet: Deep learning on point sets for 3D classification and segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, USA: IEEE.

Shi Z, et al. (2023) Deep generative models on 3D representations. Preprint, arXiv:2210.15663.

Zhang H and Blasetti E (2020) 3D architectural form style transfer through machine learning. In *Proceedings of the 25th International Conference of the Association for Computer-Aided Architectural Design Research(CAADRIA 2020)*, Hong Kong: CAADRIA.

Zhu JY, **Park T**, **Isola P and Efros AA** (2017) Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy: IEEE.