Design Space Exploration of Free-form Shells Through Conditional Variational Autoencoder

Reza Taghavifard[1], Morteza Rahbar[2]
[1]University of Montreal, LAIR [2]Iran University of Science & Technology
[1]reza.taghavifard@umontreal.ca [2]rahbarm@iust.ac.ir

In architectural design, navigating the vast solution space of free-form shells poses a major challenge due to the complexity of parametric inputs and the difficulty in evaluating structural performance early in the process. While design space exploration (DSE) is not new, existing methods often lack intuitive user interaction or neglect structural feedback in form generation. This study introduces a novel framework that combines parametric modeling, structural analysis, and a Conditional Variational Autoencoder (CVAE) to enable performance-informed exploration of free-form shells. Using three generative Grasshopper codes, 30,000 shell geometries—spanning positive, zero, and negative curvature types—were created and structurally analyzed. The CVAE learns a 16-dimensional latent space conditioned on maximum displacement under gravity. To enhance user accessibility, the space is reduced to two navigable dimensions using t-SNE, with reversibility enabled via a deep neural network. Users can intuitively explore the design space through a 2D map and generate structurally-informed shell forms with minimal input. This method supports informed decision-making by combining geometric intuition with structural performance, offering a practical tool for early-stage architectural design.

Keywords: *Design Space Exploration, Generative Design, Artificial Intelligence, Multi-Layer Perceptron Neural Network, Conditional Variational Autoencoder, Free-Form shell, Clustering.*

INTRODUCTION

Computational design has expanded the possibilities for architectural exploration, enabling the generation of vast solution sets for a given problem. This becomes especially valuable as the number of input parameters increases, allowing for the discovery of unforeseen yet high-performing outcomes. However, while these capabilities have been acknowledged for decades, efficiently navigating such high-dimensional design spaces remains a persistent challenge.

In architectural and structural design, Design Space Exploration (DSE) involves systematically analyzing a large set of design variations based on input parameters (Bird et al., 2021). Existing approaches to DSE can be broadly classified into five categories: multi-objective optimization (Pan et al., 2019; Turrin et al., 2011; Turrin et al., 2012; Yang et al., 2018), surrogate modeling (Wortmann et al., 2015; Yang et al., 2016), clustering via self-organizing maps (Pan et al., 2019; Turrin et al., 2011; Turrin et al., 2012; Wortmann et al., 2015; Yang et al., 2016; Yang et al., 2018), data visualization techniques (Harding, 2018; Wortmann and Schroepfer, 2019; Wortmann et al., 2015; Yang et al., 2016), and dimensionality reduction methods. Each technique offers a means of exploring alternatives, but few address the usability gap between algorithmic complexity and designer intuition—particularly for free-form shells.

Free-form shell structures are defined by numerous interdependent parameters and require significant upfront analysis due to their geometric and structural complexity. Achieving efficient yet expressive forms demands balancing performance constraints with architectural creativity—an enduring challenge for both designers and engineers. An automated, performance-informed tool is essential to support early-stage decision-making across such a vast design space.

Structurally, shells fall under planar systems characterized by curvature, allowing them to transfer loads through tension, compression, and shear (Farnsworth, 1999). While historically built from concrete, recent decades have seen increased use of cable nets, textile membranes, and steel grids (Adriaenssens et al., 2014). A shell's geometry directly influences its load-bearing behavior; ideally, its form should follow the natural flow of internal forces. When properly shaped, such structures minimize bending and favor membrane action, enabling thinner and more material-efficient construction.

Shell structures have historically been categorized by their geometric curvature into three types: single-curved (e.g., cylindrical), synclastic (e.g., domes), and anticlastic (e.g., hyperbolic paraboloids). This classification captures a spectrum of structural behaviors, each with distinct architectural and mechanical properties.

In recent years, machine learning and deep learning methods have increasingly informed early-stage design. Yet, many past approaches have prioritized structural efficiency over aesthetic diversity or intuitive control. The complex, high-dimensional nature of free-form shell parameter spaces remains a significant barrier to architectural adoption.

This research aims to bridge that gap by introducing a framework that enables designers to generate structurally informed shell forms through a meaningful and comprehensible latent space. These parameters are learned from data and mapped to a two-dimensional visualization, allowing interactive exploration.

Earlier methods like ParaGene by Turrin et al. (2011) integrated parametric modeling, performance simulation, and genetic algorithms, but required high upfront effort and expertise. Choo and Janssen (2015) advanced real-time interaction using surrogate models and evolutionary algorithms, addressing speed and feedback limitations. To tackle interpretability, Yang et al. (2017) combined Grasshopper and modeFRONTIER to support knowledge extraction during exploration. Wortmann (2017) proposed multivariate visualization to enhance designer intuition, though scalability to high dimensions remained limited.

Brown and Müller (2018) explored three strategies: surrogate modeling to identify variable influence, dimensionality reduction to create more intuitive control parameters, and automatic generation of input variables from geometry. However, many of these methods still struggled with generalization in large parametric spaces.

Dynamic sensitivity analysis emerged as a key technique in later works. Hinkle et al. (2024) proposed a method that dynamically evaluates the influence of variables on performance, especially suited for large, precomputed design spaces. Fang et al. (2023) extended this idea using Conditional Variational Autoencoders (CVAE) with Automatic Differentiation to both generate designs and extract mixed-variable sensitivities. In contrast to their method, our method applies CVAEs specifically to free-form shell typologies rather than mixed-variable mechanical components. While both approaches utilize latent spaces for design generation, we incorporate a 2D visualization (via t-SNE) and an inverse MLP to create a designer-accessible feedback loop. Our focus is not only on sensitivity but on intuitive and performance-aware form generation in a structurally meaningful design domain.

Fuchkina et al. (2018) introduced a flexible, hierarchical framework for DSE, combining SOM maps and clustering to enable progressive filtering of alternatives. While their system allowed storage of designer intentions, it lacked mechanisms for producing unexpected solutions. Wortman and Schroepfer [9] further surveyed performance-informed DSE, emphasizing the need for combining quantitative methods with architectural intuition via clustering, surrogate feedback, and multivariate analysis.

Pan et al. (2020) proposed an integrated method combining geometry-based clustering, neural network prediction, and interactive visualization to assist design selection. Other research applied multi-objective optimization techniques for real-time interaction and adaptive reformulation of design problems. Yang et al. [18] developed a Pareto-based, user-guided optimizer, while Chauhan et al. (2023) introduced DAISY, a real-time navigator using random forest surrogates.

Danhaive and Mueller (2021) first applied deep generative models to DSE using a CVAE and surrogate model. However, their work did not offer a visual latent space or intuitive reversibility. Mokhtar et al. (2023) used implicit neural representations to form continuous latent spaces beyond the training set, enabling semantic exploration of architectural forms. Similarly, Balmer et al. (2024) applied CVAEs to bridge design, coupling generative models with explainability tools for sensitivity analysis. Unlike their focus on mechanical typologies and explainability, our method targets free-form architectural shells and emphasizes intuitive form navigation through a bidirectionally mapped latent space.

Building on these precedents, this research introduces a CVAE-based framework for generating structurally responsive free-form shells. The system supports designers by embedding structural behavior within the latent representation and by offering an interpretable 2D design space with reversible mapping. A carefully curated dataset of free-form shells was generated across three parametric phases, covering domed, cylindrical, and hyperbolic types. Each shell was structurally analyzed to extract maximum displacement under self-weight. This informed the conditioning variable used during CVAE training.

Following a model selection study, the CVAE was adopted for its ability to interpolate between diverse forms and output structurally meaningful geometries. While the t-SNE algorithm was used to reduce the latent space for interactive exploration, its non-reversible nature was mitigated by training a separate fully connected neural network to map 2D coordinates back to design parameters. This combination enables both guided generation and exploration of shell geometries with embedded structural insight.

METHODOLOGY

Parametric Design of shells

To create a dataset for training the generative model, 30,000 shell geometries were procedurally generated using three custom parametric design algorithms in Grasshopper. Each algorithm produced 10,000 shell instances representing one of three curvature-based typologies—parabolic, elliptic, and hyperbolic—commonly used to classify architectural shells due to their distinct structural behaviors.

- Parabolic Shells (Cylindrical): Planar surfaces were subdivided using the Divide Surface command. Rectangular point sets were randomly selected and extruded vertically to form adjacent cylinders of varying heights, resulting in a shell with single curvature.

- Elliptic Shells (Domes): Using the Kangaroo physics engine, upward forces were applied to randomly selected segments of a point grid. The relaxation of these forces generated dome-like forms with consistently positive curvature across the surface.
- Hyperbolic Shells (Saddles): Also utilizing Kangaroo, this algorithm randomly selected grid points as supports and allowed the remaining points to displace freely. This produced doubly curved surfaces with negative Gaussian curvature, typical of anticlastic forms.

This taxonomy reflects structural diversity and aligns with architectural precedent. Each shell was defined within a consistent 10×10 unit domain and discretized into a uniform grid, allowing for downstream structural analysis and machine learning input standardization.

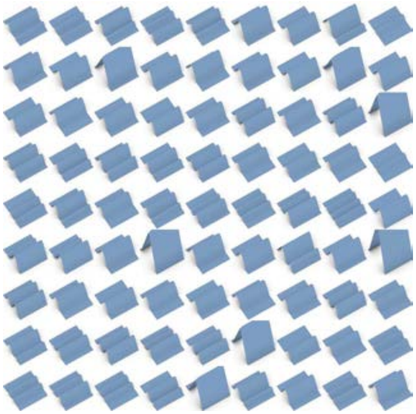

Figure 1
a set of parabolic surfaces

Dataset Preparation

Following the parametric generation of 30,000 shell geometries, each was discretized into a uniform 21×21 point grid. While the x and y coordinates were fixed across all samples, the z-values—representing vertical displacement—varied by form. For each shell, these z-values were recorded in a 21×21 matrix, serving as a compact geometric representation.

To evaluate structural performance, all shells were analyzed in Grasshopper using the Karamba plugin. The maximum displacement under self-weight was computed and stored as a scalar performance metric. This value was later used as a conditioning variable in the generative model.

To facilitate training of convolutional neural networks (CNNs) and enable easier visualization, each shell matrix was rendered as a grayscale image. The images were up sampled to 84×84 pixels, increasing spatial resolution to improve learning quality. Data augmentation via rotational symmetry was applied, expanding the dataset to 90,000 images and promoting generalization. All matrices and corresponding labels were normalized using a min-max scaler.

This dataset—comprising structured geometric inputs and scalar performance outputs—enabled training of a generative deep neural model conditioned on displacement while retaining curvature-based typological diversity. A visual summary of the pipeline is presented in Figure 4.

Conditional Variational Auto Encoder

To enable generative design exploration conditioned on structural performance, we implemented a Conditional Variational Autoencoder (CVAE) architecture incorporating convolutional neural networks (CNNs). CVAEs combine a probabilistic encoder and decoder linked by a latent space, allowing controlled generation of new samples informed by auxiliary variables—in our case, structural displacement.

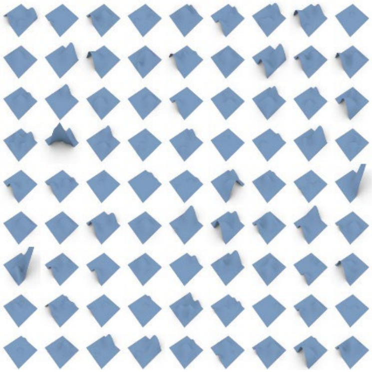

Figure 2
a set of elliptic surfaces

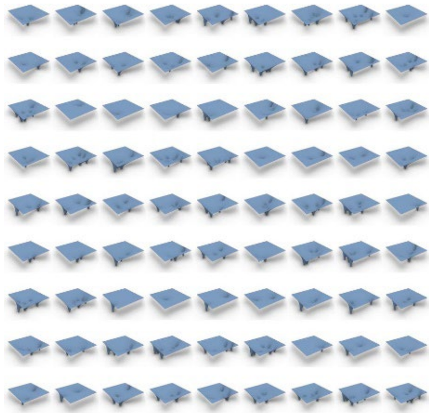

Figure 3
a set of hyperbolic surfaces

Our encoder compresses each 84×84 shell image into a 16-dimensional latent representation, capturing geometric features of the form. The decoder reconstructs images from this latent vector, extended to 17 dimensions by concatenating the maximum displacement value. This conditioning allows the model to learn structural correlations within the design space. A Kullback-Leibler divergence term in the loss function regularizes the latent distribution, encouraging generative diversity while ensuring continuity.

The encoder consists of three convolutional layers with 2×2 strides and batch normalization ($\varepsilon = 10^{-5}$, momentum = 0.1), followed by fully connected layers projecting to the latent space. The decoder mirrors this structure with three deconvolutional layers. The model is trained in batches of 64 samples. This 16D latent space serves as a compressed yet expressive manifold of shell forms, allowing for both interpolation and sampling of new geometries.

This approach differs from prior CVAE implementations (e.g., Danhaive and Mueller (2021), Balmer et al. (2024)) by integrating structural conditioning directly into the latent code and enabling two-way mapping with performance-aware generation in a domain of geometrically and mechanically distinct shell types.

Latent space dimensionality reduction

Although the CVAE reduces the design representation from 441 parameters (21×21 z-values) to 17—16 latent features plus 1 structural performance metric—this latent space remains difficult to interpret. To enable intuitive exploration and navigation, we further reduce dimensionality to a 3D space: two axes for geometric features and one for structural performance.

Principal component Analysis

As a first approach, we applied Principal Component Analysis (PCA), a linear dimensionality reduction technique that projects data onto a lower-dimensional orthogonal basis while maximizing variance (van der Maaten et al., 2008). However, PCA proved inadequate in capturing the non-linear relationships inherent in our shell geometry dataset, resulting in poor visual separability. This limitation has also been recognized in architectural applications, where PCA often struggles to retain design-relevant structure in high-dimensional parameter spaces (Harding, 2016; Serani and Diez, 2024).

t-distributed stochastic neighbor embedding

To address the limitations of PCA, we adopted t-SNE, a non-linear algorithm that preserves local similarity structures in high-dimensional data. t-SNE maps high-dimensional vectors into a 2D space where nearby points in the latent space remain close in the projection, offering a more interpretable visualization of the design manifold (Hinton and Roweis, 2002; van der Maaten and Hinton, 2008). This allowed us to meaningfully cluster and identify shells with similar forms and performance within a navigable 2D map. Its effectiveness in visualizing complex architectural design datasets has been highlighted in recent design literature (Meng, 2020).

However, t-SNE is inherently non-reversible, making it unsuitable for generative tasks. To restore generative capability, we trained a fully connected multilayer perceptron (MLP) to approximate the reverse mapping: from the 2D t-SNE space (plus performance conditioning) back to the full 17D latent code. This enables designers to select points in the 2D space and reconstruct the corresponding 3D shell geometry, achieving interactive, performance-aware design generation.

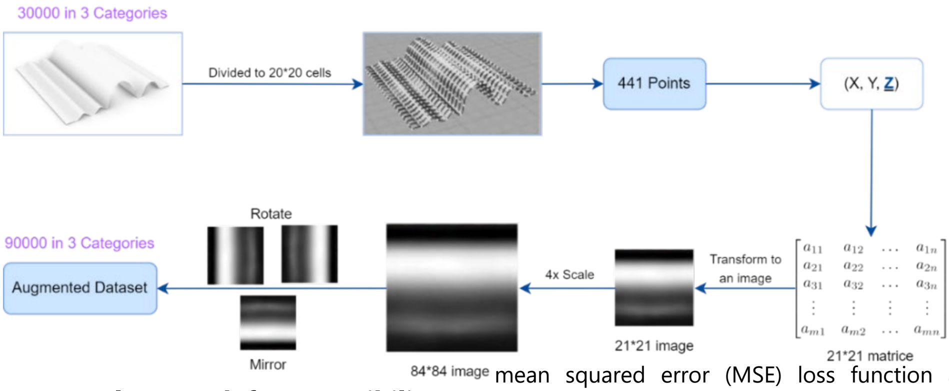

Figure 4
Data preprocessing

Deep Neural Network for reversibility

To enable generative interaction with the 2D t-SNE space, we trained a deep neural network to approximate the inverse mapping—from the 2D projection back to the original 16D latent space. This reversibility allows users to select or interpolate within the 2D space and reconstruct corresponding 3D shell geometries.

The model is a fully connected multilayer perceptron (MLP) composed of eight dense layers with a diamond-shaped architecture (see Figure 6). Input vectors of dimension 2 (plus structural performance) are processed in batches of 512. Each layer is followed by batch normalization, and ReLU is used as the activation function throughout. The network is trained using the mean squared error (MSE) loss function and optimized with the Adam optimizer (learning rate = 5×10^{-4}).

The trained MLP accurately recovered latent codes from the 2D embedding across the domain of the dataset. This capability enabled real-time design generation directly from the visualized map—making the latent space not only interpretable but also actionable for shell synthesis.

Generating Novel Shells

To synthesize new shell geometries, a user selects a point within the 2D latent space produced by t-SNE, corresponding to a specific geometric cluster (parabolic, elliptic, or hyperbolic). This point is processed by the trained

MLP to reconstruct the corresponding 16-dimensional latent vector. The structural performance value (e.g., maximum displacement) is appended to form a 17-dimensional input.

This latent-performance vector is then passed to the decoder of the CVAE, which generates a novel shell representation as an 84×84 matrix. The resulting matrix is saved in an Excel file and subsequently imported into Rhino via a Grasshopper script, where the 3D geometry is reconstructed.

This pipeline enables intuitive, performance-aware generation of novel free-form shells, allowing users to navigate the design space interactively and produce valid, structurally informed geometries without explicit parametric modeling.

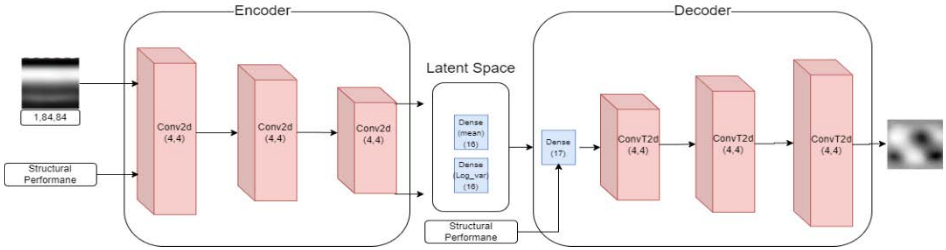

Figure 5
CVAE architecture

RESULTS AND DISCUSSION

This section presents the outcomes of the generative design framework, including an overview of the shell database, structural performance results, dimensionality reduction analysis, and examples of novel shell geometries generated through the model. Emphasis is placed on how the model integrates structural performance with geometry generation, showcasing its contribution to early-stage design

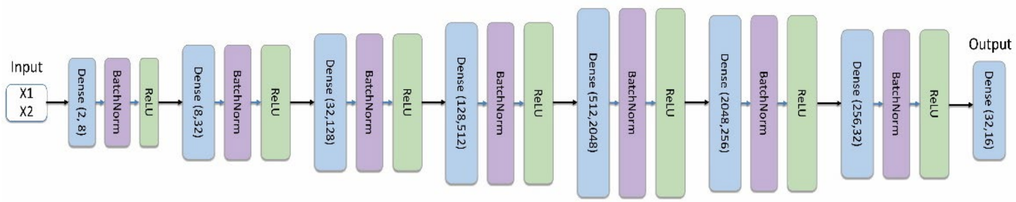

Figure 6
Deep Neural Network Architecture for the reverse process

Shell database and structural analysis

A total of 30,000 shell designs were generated across three geometric categories: domed (positive curvature), cylindrical (zero curvature in one direction), and saddle-shaped (negative curvature). All shells were evaluated for structural performance using Karamba, measuring maximum displacement under self-weight across a 10×10m span. As shown in Figure 8, hyperbolic shells consistently exhibited lower displacements, aligning with expectations due to their inherent load-distributing geometry.

Dimensionality Reduction and Latent Space Mapping

While the CVAE reduced the design input from 441 to 16 parameters, navigating this space remained non-intuitive. Principal Component Analysis (PCA) failed to meaningfully cluster different shell types in 2D. In contrast, t-SNE effectively separated the data into distinct regions corresponding to the geometric categories, as shown in Figure 9.

To enable reverse mapping from this 2D space back to the full latent space, a fully connected neural network (MLP) was trained. The MLP reached a validation MSE of 0.13, enabling accurate reconstruction of latent vectors and thereby supporting generative interaction within the 2D map.

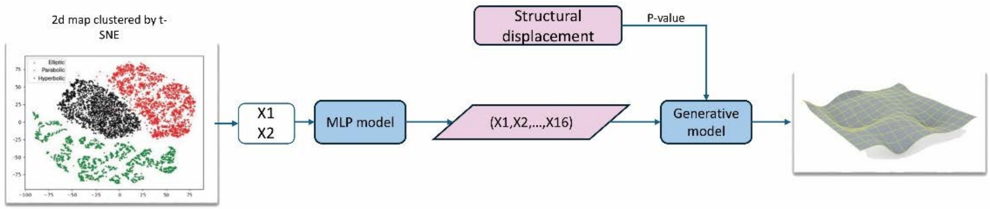

Figure 7
Generating new shells

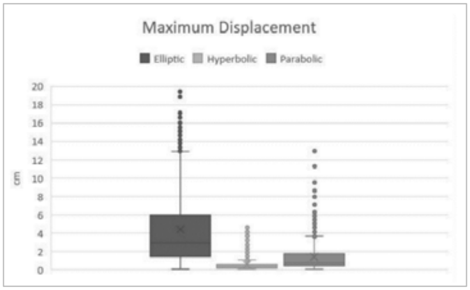

Figure 8
Range of maximum displacement obtained for all three shell groups.

Shell Generation and Performance Conditioning

Using the decoder, new shell geometries were generated from user-selected 2D positions combined with a structural performance parameter. The decoder output, an 84×84 heightmap, was imported into Rhino for 3D visualization. Generated shells often blended multiple curvature types, demonstrating the model's learned structural-spatial logic. Adjusting the performance parameter (e.g., lowering max displacement) led to geometries with increased rise and curvature (Figure 10), indicating the model's capacity to embed structural reasoning.

This illustrates a key contribution: rather than merely reconstructing training samples, the CVAE+MLP pipeline generalizes across the latent space to produce novel shells with controlled structural behavior. This contrasts prior work (e.g., Danhaive and Mueller, Balmer et al. (2024)) by introducing interactive latent conditioning, geometry-performance coupling, and reversibility in the design space.

Summary and Insights

The proposed framework simplifies early-stage design exploration by embedding structural performance into a generative model and exposing the design space through a user-navigable 2D map. It enables architects to move beyond traditional parametric modeling toward more intuitive, performance-informed decision making. While performance alone may not dictate final form, this system allows it to become a core dimension of exploration alongside spatial and aesthetic considerations.

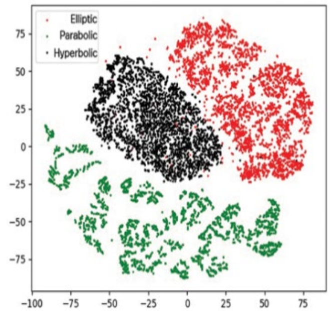

Figure 9
t-SNE performance in separation of three main categories of the shells in a 2d space

Limitations and future work

The current study is bounded by a 10×10m design domain and fixed boundary conditions. Structural performance was simplified to a single scalar value (maximum displacement), limiting nuanced assessment.

Future work can extend this framework in several directions:

- Enhanced performance feedback: Train a separate regressor to provide detailed structural indicators (e.g., stress distribution, buckling).
- Optimization integration: Use the generative model within a multi-objective optimization loop to balance performance, aesthetics, and constructability.
- Cross-domain applications: Adapt the model to other design problems such as facades, landscapes, or furniture, supporting rapid 3D generation from intuitive controls.
- Latent interpolation: Explore design variation through smooth interpolation across the latent space to support creative transitions and hybrid forms.

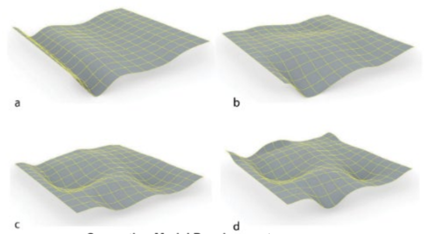

Figure 10
Effect of modifying structural performance parameter on the generated forms

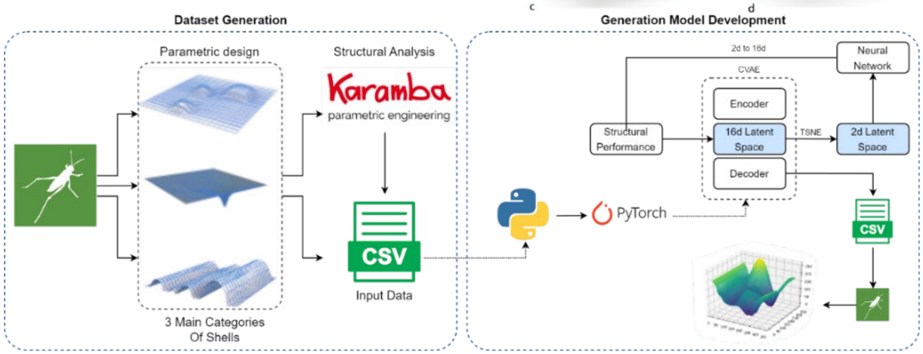

Figure 11
Overview of the research framework

REFERENCES

Balmer, V. M., Lindner, G., Ricklin, L., and Shea, K. (2024). 'Design space exploration and explanation via conditional variational autoencoders in Meta-Model-Based conceptual design of pedestrian bridges,' *Automation in Construction*, 163, p. 105411.

Bird, G. D., Gorrell, S. E., and Salmon, J. L. (2021). 'Dimensionality-reduction-based surrogate models for real-time design space exploration.' *Aerosp. Sci. Technol.*, 118, 107077.

Brown, N. C., and Mueller, C. T. (2018). 'Design variable analysis for performance-based parametric modeling.' *IJAC,* 17(1), pp. 36–52.

Chauhan, D., Orme, S., Gugliotta, S., Wynne, Z., Black-Roberts, A., and Philipps, D. P. (2023). 'Daisy: A data-driven multi-objective design tool.' *Adv. Arch. Geometry*, 111–124. .

Ding, Y., Di Stefano, D., Turrin, M., Sarıyıldız, S., and Sun, Y. (2020). 'Dynamic and interactive re-formulation of multi-objective optimization problems for conceptual architectural design exploration,' *Automation in Construction*, 118, p. 103251.

Fang, D., Kuhn, S. V., Kaufmann, W., Kraus, M. A., and Mueller, C. (2023). 'Quantifying influence of design decisions using sensitivities.' *Adv. Arch. Geometry*, 411–426.

Farnsworth, D. B. (1999). Behavior of shell structures. (MSc Thesis, MIT.)

Fuchkina, E., Schneider, S., Bertel, S., and Osineseva, I. (2018). ‘Design space exploration framework.’ Presented at the *eCAADe Conference*.
Harding, J. (2018). ‘Dimensionality reduction for parametric design exploration.’ *Adv. Arch. Geometry*.
Hinkle, L. E., Pavlak, G., Curtis, L., and Brown, N. C. (2024). ‘Dynamic subset sensitivity analysis.’ *Automation in Construction*, 158, p. 105198.
Hinton, G. E., and Roweis, S. (2002). ‘Stochastic neighbor embedding.’ *NeurIPS*, 15. pp. 857–864.
Janssen, P. (2015). ‘Performance-based parametric design framework.’ Presented at the *CAADRIA Conference*.
Maaten, L., Postma, E., and Herik, J. (2008). Dimensionality reduction: A comparative review. Technical Report.
Maaten, L., and Hinton, G. (2008). ‘Visualizing data using t-SNE.’ *JMLR*, 9(86), 2579–2605.
Meng, L. L. (2020). ‘t-SNE: A dimensionality reduction tool for design data visualisation.’ In *Proceedings of the CAADRIA Conference*.
Mokhtar, S., Danhaive, R., and Mueller, C. (2023). ‘Neural implicit fields for Performance-Informed geometries in building design.’ *Adv. Arch. Geometry*, 397–410.
Pan, W., Turrin, M., Louter, C., Sariyildiz, S., and Sun, Y. (2019). ‘Integrating multi-functional space and long-span structure in the early design stage of indoor sports arenas by using parametric modeling and multi-objective optimization.’ *J. Building Engineering*, 22, 464–485.
Pan, W., Sun, Y., Turrin, M., Louter, C., and Sariyildiz, S. (2020). ‘Design exploration of quantitative performance and geometry typology for indoor arena based on self-organizing map and multi-layered perceptron neural network.’ *Automation in Construction*, 114, p.103163.
Serani, A., and Diez, M. (2024). ‘A survey on design-space dimensionality reduction methods for shape optimization.’ *Arch. Comput. Methods Eng.*
Turrin, M., von Buelow, P., and Stouffs, R. (2011). ‘Design explorations of performance driven geometry in architectural design using parametric modeling and genetic algorithms’ *Adv. Eng. Informatics*, 25(4), 656–675.
Turrin, M., von Buelow, P., Kilian, A., and Stouffs, R. (2012) ‘Performative skins for passive comfort’ *Automation in Construction*, 22, 36-50.
Wortmann, T. (2017). 'Surveying design spaces with performance maps: A multivariate visualization method for parametric design and architectural design optimization,' *IJAC*, 15(1), pp. 38–53.
Wortmann, T., Costa, A., Nannicini, G., and Schroepfer, T. (2015). 'Advantages of surrogate models for architectural design optimization,' *AI EDAM*, 29(4), pp. 471–481.
Wortmann, T., and Schroepfer, T. (2019). ‘From Optimization to Performance-Informed Design Simulation-based Design Tools and Methods; Architectural Design Optimization; Visualization of Optimization Results’ Presented at the *SimAUD Conference*.
Yang, D., Ren, S., Turrin, M., Sariyildiz, S., and Sun, Y. (2018). 'Multi-disciplinary and multi-objective optimization problem re-formulation in computational design exploration: A case of conceptual sports building design,' *Automation in Construction*, 92, pp. 242–269.
Yang, D., Sun, Y., Di Stefano, D., and Turrin, M. (2016). 'Impacts of problem scale and sampling strategy on surrogate model accuracy: An application of surrogate-based optimization in building design,' *2022 IEEE Congress on Evolutionary Computation (CEC)*.
Yang, D., Sun, Y., Di Stefano, D., and Turrin, M. (2017). 'A Computational Design Exploration Platform Supporting the Formulation of Design Concepts,' *Proceedings of the SimAUD*.