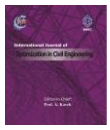
INTERNATIONAL JOURNAL OF OPTIMIZATION IN CIVIL ENGINEERING
Int. J. Optim. Civil Eng., 2025; 15(3):317-334



GENERATIVE ARTIFICIAL INTELLIGENCE IN STRUCTURAL OPTIMIZATION: OPPORTUNITIES, CHALLENGES, AND FUTURE DIRECTIONS

S. Talatahari[1,2,3*, †] and B. Nouhi[1]
[1] *Data Science Institute, Faculty of Engineering & Information Technology, University of Technology Sydney, 2007 Ultimo, Australia*
[2] *School of Computing, Macquarie University, Sydney, Australia*
[3] *Department of Computer Science, Khazar University, Baku, Azerbaijan*

ABSTRACT

The emergence of Generative Artificial Intelligence (GenAI) presents new possibilities for transforming structural optimization processes in civil and structural engineering. Unlike traditional AI models focused on prediction or classification, GenAI models, such as Generative Adversarial Networks (GANs), Variational Autoencoders (VAEs), Diffusion Models, and Large Language Models (LLMs), enable the generation of novel structural designs by learning complex patterns within design-performance data. This paper provides a comprehensive review of how GenAI can support tasks such as design generation, inverse design, data augmentation for surrogate modeling, and multi-objective trade-off exploration. It also examines key challenges, including constraint integration, model interpretability, and data scarcity. By evaluating recent applications and proposing hybrid frameworks that blend generative modeling with domain knowledge and optimization strategies, this study outlines a research roadmap for the responsible and effective use of GenAI in structural optimization. The findings emphasize the need for interdisciplinary collaboration to translate GenAI's creative potential into physically valid, structurally sound, and engineering-relevant solutions.

Keywords: Generative Artificial Intelligence, Structural Optimization, Surrogate Modeling, Variational Autoencoder, Generative Adversarial Network, Diffusion Models, Large Language Models.

Received: 25 June 2025; Accepted: 2 August 2025

---

*Corresponding author: Data Science Institute, Faculty of Engineering & Information Technology, University of Technology Sydney, 2007 Ultimo, Australia
†E-mail address: Siamak.Talatahari@mq.edu.au (S. Talatahari)

1. INTRODUCTION

The field of structural optimization has long played a pivotal role in advancing the efficiency, safety, and sustainability of civil [1-3], mechanical [4, 5], and aerospace engineering systems [6, 7]. By systematically exploring design alternatives to meet performance objectives under a range of constraints, structural optimization has enabled the development of lighter, stronger, and more cost-effective structures. Traditional approaches, ranging from gradient-based methods [8] to evolutionary algorithms [9, 10] and surrogate-assisted models [11], have significantly matured, yet they remain heavily reliant on computational resources, human intuition, and domain-specific heuristics. With growing complexity in structural systems and the demand for real-time, adaptable design solutions, there is a pressing need for more intelligent, autonomous, and generative approaches [12].

Generative Artificial Intelligence (GenAI) represents a transformative shift in how data, models, and design spaces can be synthesized and explored [13]. Unlike conventional Machine Learning (ML) models that focus primarily on prediction, GenAI encompasses a class of algorithms capable of creating new data instances that resemble the underlying distribution of the training data. Techniques such as Generative Adversarial Networks (GANs) [14], Variational Autoencoders (VAEs) [15], Diffusion models [16], and Large Language Models (LLMs) [17] have demonstrated remarkable success in fields like image generation [18], text synthesis [19], computer-aided design [20], inverse modelling [21], and simulation acceleration [22].

While recent studies have begun to explore the integration of GenAI models into structural optimization workflows, such as their application in tall building optimization [23], conceptual structural form generation [24], and surrogate modeling [25] for high-dimensional design problems, the field remains in its infancy, with most developments still at a proof-of-concept or exploratory stage. Importantly, most GenAI models were originally developed for domains like image, text, and data synthesis, and are not inherently tailored to meet the specific constraints and physical laws governing structural systems, [26]. As a result, their direct application in structural optimization presents several challenges.

This paper aims to provide a comprehensive overview of the intersection between GenAI and structural optimization. We begin by outlining the foundational principles of both fields, followed by an examination of the current research landscape that explores their intersection and integration. Key opportunities and challenges are identified, followed by a discussion on emerging applications and future directions. By synthesizing knowledge across computational intelligence and structural engineering, this paper seeks to guide researchers and practitioners in leveraging GenAI to shape the next generation of intelligent design and optimization systems.

2. BACKGROUND

2.1. Structural Optimization

Structural optimization focuses on systematically improving a structure's performance while satisfying design constraints. It is typically formulated as, [2]:

$$\begin{aligned} &\min_{x} f(\boldsymbol{x}) \\ s.t.\quad & g_i(\boldsymbol{x}) \le 0, \qquad i = 1,2,\dots,m \\ & h_j(\boldsymbol{x}) = 0, \qquad j = 1,2,\dots,p \\ & \boldsymbol{x} \in \boldsymbol{\Omega} \end{aligned} \tag{1}$$

where $\boldsymbol{x}$ denotes the design variables (number of sections or cross-sectional area of elements or geometry variables), $f(x)$ is the objective function (e.g., weight or cost), and $g_i(x)$ and $h_j(x)$ are inequality and equality constraints (e.g., stress, displacement, frequency or geometry constraints).

Depending on the representation of design variables and objectives, problems are classified as:

a) Size Optimization

In size optimization, the structural layout or topology is assumed to be fixed, and the design process focuses on finding the optimal values of specific dimensional parameters that influence performance, [27]. The design variables typically correspond to the sizes of structural components, such as cross-sectional areas of truss or frame members, thicknesses of plates or shells, or diameters of reinforcing bars.

The objective of size optimization is often to minimize structural weight, material cost, or compliance (a measure of structural flexibility), while satisfying stress, displacement, buckling, frequency, and serviceability constraints. A general formulation can be written as, [1]:

$$\begin{aligned} &\min_{x} f(\boldsymbol{x}) = \sum_{i=1}^{n} \rho l_i x_i \\ s.t.\quad & \sigma_i(\boldsymbol{x}) \le \sigma_{allow}, \ \forall i \\ & \delta_j(\boldsymbol{x}) \le \delta_{allow}, \ \forall j \\ & x_i^{min} \le x_i \le x_i^{max}, \ i = 1,2,\dots,n \end{aligned} \tag{2}$$

where, x_i are the size variables, ρ is the material density, l_i is the element length, σ_i and δ_j are stress and displacement values, σ_{allow} and δ_{max} are allowable stress and displacement limits.

Size optimization is widely used in preliminary and detailed design stages for steel frames, trusses, reinforced concrete members, and composite structures. It is also a core component of many code-based optimization routines, where standard section sizes must be selected from design tables to meet structural requirements.

b) Shape Optimization

In shape optimization, the goal is to enhance structural performance by modifying the geometry of the structure's boundaries, such as the outer contour or internal voids, while keeping the topology (i.e., connectivity and number of components) fixed, [28]. This type of optimization allows adjustment of both the positions of boundary nodes and control parameters of geometric entities. In some formulations, shape optimization is performed in combination with size optimization, enabling simultaneous refinement of geometric boundaries and element dimensions. Unlike size optimization, which adjusts scalar design variables associated with element properties, shape optimization directly affects the spatial configuration of the structure. However, it does not allow the creation or removal of holes, branches, or components, as that would alter the topology.

The mathematical formulation of shape optimization is often embedded within a structural analysis framework, and can be written as, [29]:

$$\begin{gathered} \min_{\boldsymbol{x}_{shape}} f(\boldsymbol{x}_{shape}) \\ s.t. \quad \mathbf{K}(\boldsymbol{x}_{shape})\boldsymbol{u} = \boldsymbol{F} \\ s.t. \quad g_i(\boldsymbol{x}_{shape}) \leq 0, \qquad i = 1,2,\dots,m \\ \boldsymbol{x}_{shape} \in \boldsymbol{\Omega}_{admissible} \end{gathered} \tag{3}$$

where, $\boldsymbol{x}_{shape}$ denotes the coordinates or shape parameters, $f(\boldsymbol{x}_{shape})$ is the objective function, $\mathbf{K}(\boldsymbol{x}_{shape})$ is the shape-dependent stiffness matrix, $\boldsymbol{u}$ is the displacement vector obtained from finite element analysis (FEA), and $\boldsymbol{F}$ is the external force vector. $g_i(\cdot)$ are performance or constraint functions, and $\boldsymbol{\Omega}_{\mathbf{admissible}}$ defines bounds on shape parameters to ensure manufacturability or physical feasibility.

This problem is widely used in aerospace, mechanical, and civil engineering applications where precise control over stress concentration zones, flow boundaries, or resonance frequencies is critical.

c) Topology Optimization

Topology optimization seeks to determine the optimal material layout within a predefined design domain to achieve the best structural performance under given loading and boundary conditions, [30]. Unlike size or shape optimization, which operate within a predefined structural configuration, topology optimization does not assume any initial layout. It allows for the emergence of new holes, branches, and load paths, enabling innovative, highly efficient, and often unintuitive designs.

The design variable in topology optimization is typically a material density field $\rho(x)\in[0,1]$, defined over the elements or voxels of the discretized domain. A density of $\rho=1$ indicates the presence of solid material, while $\rho=0$ represents void. Intermediate values are penalized to push the solution toward a binary (0–1) distribution.

The standard compliance minimization problem in topology optimization can be formulated as, [31, 32]:

$$\begin{gathered} \min_{\rho} f(\rho) = \boldsymbol{u}^T \mathbf{K}(\rho)\boldsymbol{u} \\ s.t. \quad V(\rho) = \sum_e \rho_i v_e \ \leq V_0 \\ \mathbf{K}(\rho)\boldsymbol{u} = \boldsymbol{F}, \ \ \rho_e \in [\rho_{min}, 1] \ \forall e \end{gathered} \tag{4}$$

where, v_e is the volume of element e, V_0 is the maximum allowable material volume, and ρ_{min} is a small positive lower bound to avoid singular stiffness matrices.

Topology optimization is computationally intensive because it involves solving a finite element equilibrium equation and updating material distribution iteratively. Despite these costs, it has become an essential tool in lightweight design, additive manufacturing, and conceptual design of structural systems in aerospace, automotive, and civil engineering domains.

2.2. Solution Strategies in Structural Optimization

Selecting an appropriate optimization strategy is critical for solving structural optimization problems efficiently and reliably. The choice largely depends on the nature of the objective and constraint functions, the size and dimensionality of the problem, and the availability of gradient information. Broadly, solution strategies can be categorized into gradient-based methods, metaheuristic algorithms, surrogate-assisted methods, and hybrid frameworks that integrate multiple paradigms (Figure 1).

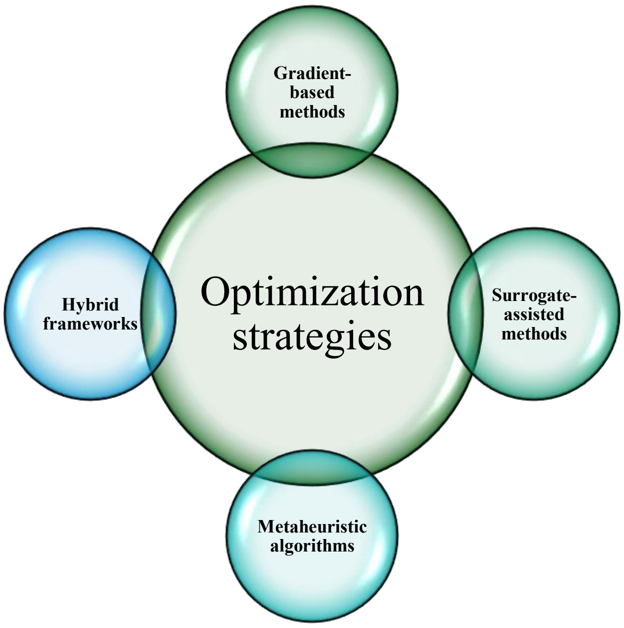


Figure 1: Optimization strategies for solving structural optimization.

Gradient-based optimization methods rely on first- and second-order derivative information to guide the search direction. These methods, such as Sequential Quadratic Programming (SQP) [33], Method of Moving Asymptotes (MMA) [34], and Interior Point Methods [35], are particularly effective for problems with smooth, differentiable objective and constraint functions, as commonly encountered in size and shape optimization with well-behaved finite element models. Their convergence is typically fast and efficient, especially in low- to moderate-dimensional spaces. However, they are sensitive to local minima, may fail in non-convex or discrete spaces, and require accurate sensitivity analysis, which can be computationally expensive or difficult to derive analytically for complex systems.

To address non-convexity, discontinuities, and large-scale combinatorial spaces, particularly in large-scale design optimization problems, material layout problems, or constraint-rich design scenarios, metaheuristic algorithms are widely employed, [36]. These methods do not require gradient information and are well-suited for exploring complex design spaces. However, their stochastic nature often leads to longer convergence times, and they may require a large number of objective function evaluations, posing a significant burden when each evaluation involves expensive simulations, [12].

To mitigate computational cost, especially in high-fidelity or high-dimensional problems, surrogate-assisted optimization has become a prominent strategy, [37]. Surrogate models use inexpensive approximations to estimate the objective and constraint functions, thereby reducing the number of expensive evaluations. Common surrogate modeling techniques include Gaussian Process Regression [38], Radial Basis Function [39], Polynomial Regression [40], Artificial Neural Networks [41] along with other traditional ML approaches widely used for function approximation in structural optimization. In such frameworks, an initial sample set is used to train the surrogate model, which is then refined iteratively via adaptive sampling, infill criteria, and uncertainty quantification.

Recently, hybrid frameworks have gained attention by integrating surrogate models, metaheuristics [42], and even Reinforcement Learning [43] to dynamically balance exploration and exploitation, adapt search strategies on-the-fly, and handle expensive or partially known objective landscapes. Such frameworks are particularly suitable for multi-objective optimization [44], expensive problems [45], and real-time applications in structural health monitoring [46].

3. FROM ARTIFICIAL INTELLIGENCE TO GENAI: PRINCIPLES AND TECHNIQUES

Artificial Intelligence (AI) is a foundational discipline in computer science that aims to develop computational systems capable of mimicking human cognitive functions such as reasoning, learning, perception, decision-making, and adaptation. Historically, AI began as a rule-based system paradigm, where explicit logical rules and symbolic reasoning engines were manually encoded to solve problems, [47]. Early AI applications included chess-playing programs, expert systems for medical diagnosis, and search-based planning algorithms. While powerful for well-defined, narrow domains, these systems struggled with ambiguity, scale, and the variability inherent in real-world data, [48].

The field evolved with the rise of ML, a subset of AI that allows systems to learn from data rather than rely on hard-coded rules. ML models infer patterns, trends, and relationships within datasets to make predictions or decisions without being explicitly programmed for every possible scenario, [49]. ML is generally divided into, [50]:

- **Supervised learning**, where models learn from labeled,
- **Unsupervised learning**, which discovers hidden structures in unlabeled data,
- **Reinforcement learning**, where agents learn to make sequences of decisions through reward-based feedback.

Most classical ML models are discriminative, meaning they aim to model the conditional probability $P(y|\boldsymbol{x})$, learning to map inputs $\boldsymbol{x}$ to outputs y. These models have powered a wide array of applications, including predictive maintenance, structural health monitoring, and automated quality control.

However, GenAI represents a paradigm shift within ML. Instead of predicting labels or outcomes, GenAI models learn the joint probability distribution $P(\boldsymbol{x})$ or conditional generative processes $P(\boldsymbol{x}|z)$, where z is a latent variable, [13]. In doing so, these models can generate new data instances that are statistically similar to those seen during training, often exhibiting remarkable creativity, diversity, and realism. This capability allows GenAI to answer fundamentally different questions, "What could a plausible design look like?" or "What is the space of feasible structural configurations?".

This transition has been incremental but transformative. Early generative models like naive Bayes and Hidden Markov Models (HMMs) [51] laid the groundwork for probabilistic generation in simple domains. Subsequent advances in neural networks have led to the development of powerful deep generative models, including techniques such as Variational Autoencoders, Generative Adversarial Networks, Diffusion Models, and Transformer-based architectures.

a) Generative Adversarial Networks

GANs are a class of generative models composed of two neural networks, the generator $G(z)$ and the discriminator $D(\boldsymbol{x})$, trained simultaneously in a competitive setting as shown in Figure 2, [14]. The generator maps a latent vector $\boldsymbol{z}$, typically sampled from a multivariate normal distribution $N(0,\mathrm{I})$, to the data space to produce synthetic samples $\hat{\boldsymbol{x}} = G(z)$. The discriminator receives both real samples $\boldsymbol{x} \sim P_{data}$ and generated ones, and learns to distinguish between the two, outputting a probability that indicates whether a given input is real or fake. The training process is formulated as a minimax game, in which the discriminator seeks to maximize its ability to correctly distinguish real from generated data, while the generator simultaneously attempts to minimize this ability by producing increasingly realistic outputs, [52]. Mathematically, this adversarial objective is expressed as, [14]:

$$\min_G \max_D \mathbb{E}_{\boldsymbol{x}\sim P_{data}}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim P_z}[\log(1 - D(G(\boldsymbol{x})))] \tag{5}$$

GANs have demonstrated strong capabilities in producing high-fidelity, high-dimensional outputs, making them attractive for generative tasks in many fields. However, they also face significant challenges including training instability, mode collapse, and sensitivity to network architecture and hyperparameters. Despite these limitations, their capacity to learn implicit distributions without explicit labels makes GANs a promising tool in early-stage structural design and conceptual modeling workflows.

```
Initialize Generator G and Discriminator D with random weights
for number of training epochs:
    for each batch of real data x_real:

        # ---- Train Discriminator ----
        Sample noise vector z from prior distribution (e.g., normal)
```

```
        Generate fake data: x_fake = G(z)

        Compute D_loss:
           - D_loss_real = -log(D(x_real))
           - D_loss_fake = -log(1 - D(x_fake))
           - D_loss = D_loss_real + D_loss_fake

        Update Discriminator parameters to minimize D_loss

        # ---- Train Generator ----
        Sample new noise vector z
        Generate fake data: x_fake = G(z)

        Compute G_loss:
           - G_loss = -log(D(x_fake))  # Generator tries to fool the discriminator

        Update Generator parameters to minimize G_loss
```

Figure 2: Simplified pseudocode of the GAN training loop.

b) Variational Autoencoders

VAEs are a class of probabilistic generative models that learn to represent input data in a compressed latent space and generate new, similar data by sampling from that latent space, [15]. A VAE consists of two main components: an encoder and a decoder. Figure 3 provide the pseudocode of the VAE training loop. The encoder maps an input $\boldsymbol{x}$ to a distribution over a latent variable $\boldsymbol{z}$, typically a multivariate Gaussian $q_\phi(\boldsymbol{z}|\boldsymbol{x})$~$N(\mu,\sigma^2)$. The decoder then samples from this latent space and reconstructs the data via $p_\theta(\boldsymbol{x}|\boldsymbol{z})$. Unlike classical autoencoders, which learn deterministic mappings, VAEs incorporate stochasticity and are trained to maximize the Evidence Lower Bound (ELBO) on the data likelihood, [53].

The training objective of a VAE balances two goals: (1) ensuring accurate reconstruction of the input data and (2) regularizing the latent space to follow a prior distribution, typically $p(z)=N(0,I)$. The loss function is given by, [15]:

$$\mathcal{L}(\theta,\emptyset;\boldsymbol{x}) = \mathbb{E}_{q_\emptyset(\boldsymbol{z}|\boldsymbol{x})}[\log(P_\theta(\boldsymbol{x}|\boldsymbol{z}))] - \mathrm{KL}\,[q_\emptyset(\boldsymbol{z}|\boldsymbol{x})\|P(\boldsymbol{z})] \tag{6}$$

where, $q_\phi(\boldsymbol{z}|\boldsymbol{x})$ is the encoder, $p_\theta(\boldsymbol{x}|\boldsymbol{z})$ is the decoder (generative model) and KL[·‖·] is the Kullback–Leibler divergence, enforcing similarity between the learned latent distribution and the prior.

VAEs ability to produce diverse yet plausible outputs, while maintaining a smooth and interpretable latent space, makes them attractive for tasks such as parametric form generation, topology exploration, and embedding structural constraints into generative processes.

```
Initialize Encoder network qϕ(z|x)
Initialize Decoder network pθ(x|z)

for number of training epochs:
    for each batch of input data x:

        # ---- Encode ----
        Encode x to obtain mean μ and standard deviation σ
        Sample latent vector z using reparameterization:
           z = μ + σ * ε, where ε ~ N(0, I)
```

```
    # ---- Decode ----
    Reconstruct input: x_recon = Decoder(z)

    # ---- Compute Loss ----
    Compute reconstruction loss:
        L_recon = ||x - x_recon||^2  (or use binary cross-entropy)

    Compute KL divergence loss:
        L_KL = D_KL[ N(μ, σ^2) || N(0, I) ]

    Total loss:
        L_total = L_recon + L_KL

    # ---- Update Parameters ----
    Backpropagate and update Encoder and Decoder to minimize L_total
```

Figure 3: Simplified pseudocode of the VAE training loop.

c) Diffusion Models

Diffusion Models are a class of generative models that learn to generate data by reversing a gradual noising process, [16]. The core idea is to start with a sample of pure noise and progressively denoise it over a series of steps to produce a coherent, realistic data sample. This is achieved by first training a model to learn the reverse of a diffusion (noising) process, which systematically corrupts data over time by adding small amounts of Gaussian noise. During training, the model learns to estimate the denoised data at each time step, allowing it to reconstruct data from noise at inference time.

Mathematically, the forward diffusion process gradually transforms a data sample $\boldsymbol{x}_0$ into a noise vector $\boldsymbol{x}_T$ through a series of steps, [16]:

$$q(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}) = N(\boldsymbol{x}_t; \sqrt{1-\beta_t}\boldsymbol{x}_{t-1}, \beta_t\boldsymbol{I}) \tag{7}$$

where β_t controls the noise variance at each timestep. The generative model then learns the reverse process $p_\theta(\boldsymbol{x}_{t-1}|\boldsymbol{x}_t)$, effectively denoising $\boldsymbol{x}_t$ back to $\boldsymbol{x}_0$. Simplified pseudocode of the training loop for a Diffusion Model is presented in Figure 4.

High output diversity, stable training dynamics, and strong fidelity to the underlying data distribution are among the key advantages of diffusion models. However, these benefits come at the cost of increased computational complexity, long inference times, and the need for large training datasets, which can pose challenges for their deployment in real-time or resource-constrained structural optimization workflows.

```
Initialize denoising neural network εθ(x_t, t)

for number of training epochs:
    for each batch of real data x_0:

        # ---- Forward diffusion (noising process) ----
        Sample timestep t ~ Uniform(1, T)
        Sample noise ε ~ N(0, I)
        Compute noisy input:
            x_t = sqrt(ᾱ_t) * x_0 + sqrt(1 - ᾱ_t) * ε

        # ---- Training objective ----
        Predict noise: ε_pred = εθ(x_t, t)
        Compute loss:
```

```
        L = ||ε - ε_pred||^2

    # ---- Update model parameters ----
    Backpropagate and update εθ to minimize L
```

Figure 4: Simplified pseudocode of the training loop for a Diffusion Model.

d) Large Language Models

LLMs are transformer-based deep learning models trained to understand and generate human-like text by modeling the probability distribution over sequences of tokens, [54]. These models, such as GPT, BERT, and LLaMA, are trained on massive corpora of text and learn to capture complex semantic, syntactic, and contextual relationships. Given a sequence of tokens (x_1,x_2,…,x_T), LLMs model the joint distribution as a product of conditional probabilities, [17]:

$$P(x_1, x_2, \dots, x_T) = \prod_{t=1}^{T} P(x_t | x_1, x_2, \dots, x_{t-1}) \tag{8}$$

This autoregressive formulation allows them to perform a wide range of tasks such as code generation, symbolic reasoning, knowledge retrieval, and question answering—without task-specific retraining. Simplified pseudocode of an autoregressive LLM training loop is as provided in Figure 5.

Although initially developed for language-based tasks, LLMs are increasingly being applied in engineering domains due to their ability to interface with structured knowledge, generate design-related code or scripts, and translate high-level specifications into computational tasks.

```
Initialize Transformer-based model with parameters θ

for number of training epochs:
    for each batch of token sequences (x₁, x₂, ..., x_T):

        # ---- Language Modeling Objective ----
        For each position t in the sequence:
            Predict next token: x̂_t = Model(x₁, ..., x_{t-1})

        # ---- Compute Loss ----
        Cross-entropy loss:
            L = - Σ_t log P(x_t | x₁, ..., x_{t-1})

        # ---- Update Model ----
        Backpropagate and update θ to minimize L
```

Figure 5: Simplified pseudocode of an autoregressive LLM training loop.

4. INTERSECTION OF GENAI AND STRUCTURAL OPTIMIZATION

The integration of GenAI into structural optimization introduces a transformative shift in how design spaces are explored, evaluated, and automated. While traditional optimization workflows rely heavily on numerical simulations, heuristic search algorithms, or manually constructed surrogate models, GenAI offers new capabilities such as the automated generation of structurally plausible forms, inversion of performance-to-design mappings, and intelligent support for navigating complex multi-objective trade-offs.

Figure 6 presents a bibliometric co-occurrence network generated from Scopus [55], using keywords related to generative models and structural optimization. The visualization reveals several thematic clusters, where the size of each node represents keyword frequency, and the thickness of edges indicates co-occurrence strength. As observed, "deep learning" emerges as the most central and frequently occurring term, reflecting its dominant role in current AI-driven structural optimization research. Closely linked terms such as "topology optimization", and "structural design" form a core cluster of applied methodologies.

In contrast, terms explicitly associated with "generative AI", such as "generative adversarial networks", "variational autoencoder", "diffusion model", and "transformer", appear with much lower frequency and are situated on the periphery of the network. This suggests that while deep learning methods are well established in the structural optimization domain, the application of GeneAI techniques remains emergent and exploratory. The presence of clusters like "generative design", "intelligent structural design", and "design automation" indicates a growing but still fragmented research effort toward integrating GenAI into structural workflows. Overall, the figure highlights both the increasing interest in this interdisciplinary space and the substantial opportunity for deeper integration of generative AI into structural design and optimization research.

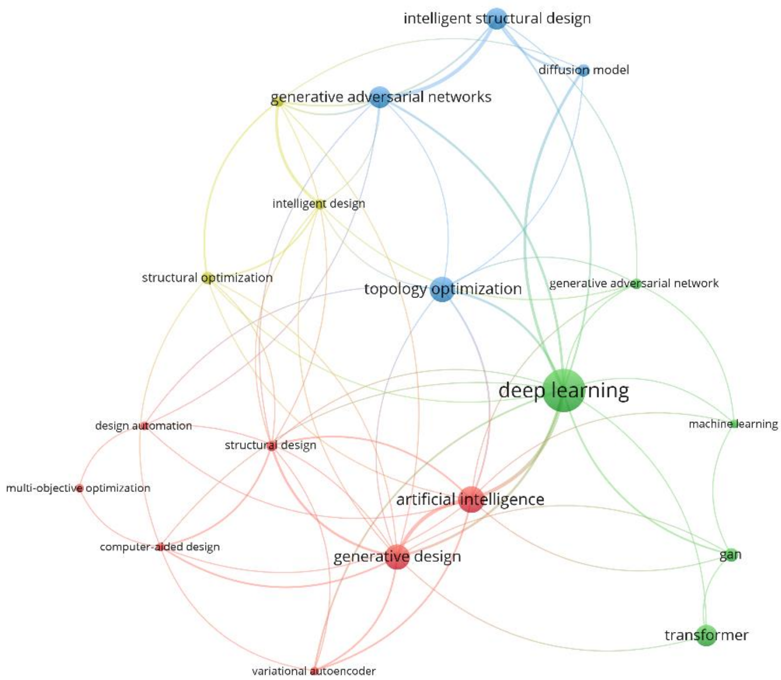


Figure 6: Keyword co-occurrence network related to generative AI and structural optimization, based on Scopus-indexed publications, (based on the data of [55]).

4.1. Design Generation

One of the most direct applications of GenAI in structural optimization is design generation, where generative models are trained to produce plausible structural layouts, geometries, or topologies. Given a dataset of optimized or physically valid designs, these models can learn the underlying distribution and generate new design candidates that adhere to similar structural principles. This approach is particularly valuable in early-stage conceptual design, where engineers often require a diverse set of creative yet structurally viable alternatives without executing a full-scale optimization process for each option, [23]. For instance, GANs have been applied to generate initial shear wall configurations for tall buildings, providing a foundation for downstream refinement through optimization techniques [56, 57]. Such use cases demonstrate the ability of GenAI to accelerate ideation, enable automation of preliminary layouts, and reduce the computational burden in design iteration cycles, {Gradišar, 2024 #58}.

4.2. Data Augmentation for Training Surrogates

In many structural optimization scenarios, building accurate surrogate models is limited by the scarcity of diverse, high-quality training data, especially when obtaining design samples through physics-based simulations is computationally expensive, {Talatahari, 2025 #12}. Generative AI models offer a powerful mechanism for data augmentation by learning compact, meaningful representations of design spaces and generating new, performance-informed design instances.

Recent studies {Danhaive, 2021 #59} have demonstrated how design subspace learning using VAEs can uncover a structured latent space that captures the essential features of feasible designs. From this latent space, new candidate designs can be sampled that preserve structural plausibility and performance similarity to known solutions. For example, a VAE trained on a limited dataset of optimized structure configurations can generate diverse variants that conform to learned engineering constraints, thereby expanding the training set for surrogate models without requiring additional full-scale simulations. This GenAI-driven data generation not only improves diversity and coverage of the design space but also reduces overfitting and enhances generalization of surrogate models, especially in high-dimensional or multi-fidelity optimization settings. Furthermore, integrating generated data into surrogate-assisted workflows provides a performance-informed yet creative design augmentation strategy, overcoming traditional bottlenecks in data-limited structural optimization pipelines.

4.3. Multi-Objective Trade-Off Exploration

Structural optimization frequently involves multiple, often conflicting objectives, such as minimizing weight while maximizing stiffness, or reducing cost without compromising safety, {TalatAhari, 2024 #61}. Exploring these trade-offs requires not only efficient sampling of the design space but also intelligent strategies for constructing and navigating the Pareto front. Generative AI models, when integrated with multi-objective optimization frameworks, provide a promising avenue for enhancing this exploration.

The recent use of tuned constraint-modified GANs for data augmentation in complex material design (as shown in engineered cementitious composite mixture optimization studies) demonstrates the potential of generative models to expand feasible regions of the design space and support multi-objective search, {Wang, 2024 #62}. Optimization algorithms, which incorporate reference-point strategies, can then efficiently guide the search through high-dimensional design spaces involving both performance and cost objectives.

4.4. Encoding Design Constraints and Objectives

A major challenge in generative design is ensuring that generated outputs satisfy domain-specific constraints, such as stress limits, deflection bounds, or regulatory codes. Traditional GenAI models operate in unconstrained domains, but recent efforts have explored ways to embed constraints explicitly or implicitly. One avenue is through prompt engineering or fine-tuning LLMs to interpret constraints described in engineering language and translate them into structured input for optimization or simulation tools, {Ploennigs, 2024 #63}. Alternatively, physical constraints can be embedded during training through loss function penalties or physics-informed architectures to bias generation toward feasible designs.

5. OPPORTUNITIES AND CHALLENGES

The integration of GenAI into structural optimization presents a unique opportunity to redefine how engineers approach design exploration, modeling, and problem-solving. While early-stage studies have begun to reveal its transformative potential, substantial gaps remain in theoretical development, practical application, and integration with existing engineering tools. This section presents a balanced assessment of the major opportunities offered by GenAI in structural optimization, along with the key challenges that must be addressed to enable wider adoption and impact.

The benefits of using GenAI in this field can be summarised as follows:

a) Accelerated and Automated Design Exploration: GenAI enables the rapid generation of structurally plausible designs, allowing engineers to explore large, complex design spaces without solving full-scale optimization problems repeatedly. This is particularly beneficial in early-stage conceptual design or for ideation across multi-objective trade-offs.

b) Learning from Prior Knowledge and Data: Unlike traditional optimization methods that start from scratch, generative models can learn structural patterns from prior designs or simulation datasets. This enables knowledge reuse, data-driven creativity, and the ability to generalize across similar structural problems or geometries.

c) Bridging Engineering and Natural Language Interfaces: LLMs allow engineers to articulate constraints, objectives, or preferences using natural language, lowering the barrier for non-expert users. This paves the way for interactive design assistants and generative co-pilots that can translate engineering intent into computational design parameters.

d) Integration with Surrogates and Reduced-Order Models: Generative models complement surrogate modeling by enriching training datasets and representing design distributions in latent space. They can also produce inputs to surrogate models, thereby reducing the cost of expensive simulations and supporting real-time design iterations.

e) Enabling Inverse and Constraint-Aware Design: Conditional generative models allow for inverse mapping from desired structural performance to feasible design candidates, bypassing the need for iterative trial-and-error. Emerging physics-informed GenAI approaches can further integrate physical constraints directly into the generative process.

Despite these opportunities, several challenges must also be acknowledged and addressed to ensure the effective integration of Generative AI into structural optimization, such as:

a) Lack of Domain-Specific Training Data: High-performing generative models require large, diverse datasets, but such data is limited in civil and structural engineering. Unlike domains like computer vision, curated databases of labeled structural layouts, simulations, or design-performance pairs are scarce.

b) Physical Inconsistency and Constraint Violation: Most GenAI models were developed for media domains and do not inherently enforce equilibrium, compatibility, or material laws. Without explicit physics integration, generated designs may be infeasible or violate structural safety requirements.

c) Model Interpretability and Trust: Engineering applications demand interpretability and traceability, especially when safety is involved. However, the latent representations learned by generative models are often difficult to interpret, making it challenging to justify decisions or understand why a particular design was generated.

d) Integration Complexity: Embedding generative models into existing structural workflows, especially those involving FEA solvers, BIM systems, or CAD software, requires significant effort in interface development, data translation, and validation. Real-time interaction remains limited by computational overheads.

e) Limited Adoption and Research Maturity: Despite growing interest, the application of GenAI in structural optimization remains in its infancy. Most studies are exploratory, with few benchmark datasets, no standardized evaluation metrics, and limited replication or industrial deployment.

6. FUTURE DIRECTIONS

The integration of Generative AI into structural optimization is still in its early stages, and future research must address several technical, theoretical, and application-oriented directions to unlock its full potential. One promising avenue lies in developing domain-adapted generative architectures that explicitly incorporate physics-based constraints, structural mechanics knowledge, and regulatory design codes. This would allow generative models to move beyond data-driven plausibility and toward engineering-validity, reducing the reliance on post-processing or simulation-based validation.

Moreover, advancing hybrid frameworks that couple GenAI with traditional optimization solvers, surrogate models, or reinforcement learning agents can improve scalability and generalization across a wider range of structural problems, from early-stage conceptual design to detailed component-level optimization. The use of multimodal inputs (such as combining textual prompts, sketches, and load conditions) and human-in-the-loop systems could further support collaborative design exploration, enabling engineers to interactively steer generative processes based on both quantitative performance and qualitative preferences.

Lastly, future studies should emphasize benchmarking, interpretability, and trustworthiness of GenAI-generated designs. Establishing standard datasets, evaluation protocols, and open-source frameworks will be crucial for reproducibility and adoption in engineering practice. Additionally, ethical considerations, such as the risk of generating infeasible or unsafe designs, must be addressed through transparency, validation pipelines, and uncertainty quantification. These directions will shape a more robust and reliable future for GenAI-assisted structural optimization.

7. CONCLUSION

This study has reviewed the emerging intersection between Generative Artificial Intelligence and structural optimization, highlighting the opportunities, methodologies, and challenges involved. Generative models such as VAEs, GANs, diffusion models, and LLMs offer promising capabilities for design generation, inverse modeling, surrogate data augmentation, and multi-objective trade-off exploration. While early applications demonstrate potential in structural form discovery and performance-informed design, significant challenges remain, particularly in ensuring physical validity, interpretability, and integration with existing engineering workflows. As research advances, combining domain knowledge with generative architectures and hybrid optimization frameworks will be essential to fully realize the transformative potential of GenAI in structural design.

REFERENCES

1. Yang XS, Talatahari S, Alavi AH. *Metaheuristic Applications in Structures and Infrastructures*. Newnes; 2013.
2. Kaveh A. *Applications of Metaheuristic Optimization Algorithms in Civil Engineering*. Springer; 2017.
3. Kaveh A, Eslamlou AD. *Metaheuristic Optimization Algorithms in Civil Engineering: New Applications*. Springer; 2020.
4. Abd Elaziz M, et al. Advanced metaheuristic techniques for mechanical design problems. *Arch Comput Methods Eng*. 2021:1–22.
5. Pereira JLJ, et al. A review of multi-objective optimization: methods and algorithms in mechanical engineering problems. *Arch Comput Methods Eng*. 2022;**29**(4):2285–308.
6. Arias-Montano A, Coello CAC, Mezura-Montes E. Multiobjective evolutionary algorithms in aeronautical and aerospace engineering. *IEEE Trans Evol Comput*. 2012;**16**(5):662–94.
7. Hassan K, et al. Application of artificial intelligence in aerospace engineering and its future directions: a systematic quantitative literature review. *Arch Comput Methods Eng*. 2024;**31**(7):4031–86.

8. Kegl M, Butinar B, Kegl B. An efficient gradient-based optimization algorithm for mechanical systems. *Commun Numer Methods Eng*. 2002;**18**(5):363–71.
9. Hare W, Nutini J, Tesfamariam S. A survey of non-gradient optimization methods in structural engineering. *Adv Eng Softw*. 2013;**59**:19–28.
10. Munk DJ, Vio GA, Steven GP. Topology and shape optimization methods using evolutionary algorithms: a review. *Struct Multidiscip Optim*. 2015;**52**(3):613–31.
11. Samadian D, Muhit IB, Dawood N. Application of data-driven surrogate models in structural engineering: a literature review. *Arch Comput Methods Eng*. 2024:1–50.
12. Talatahari S, Chen F, Gandomi AH. Developing a robust machine learning framework for predicting the behavior of large-scale structure. *J Build Eng*. 2025;**105**:112204.
13. Sengar SS, et al. Generative artificial intelligence: a systematic review and applications. *Multimed Tools Appl*. 2024:1–40.
14. Goodfellow I, et al. Generative adversarial networks. *Commun ACM*. 2020;**63**(11):139–44.
15. Kingma DP, Welling M. An introduction to variational autoencoders. *Found Trends Mach Learn*. 2019;**12**(4):307–92.
16. Yang L, et al. Diffusion models: A comprehensive survey of methods and applications. *ACM Comput Surv*. 2023;**56**(4):1–39.
17. Chang Y, et al. A survey on evaluation of large language models. *ACM Trans Intell Syst Technol*. 2024;**15**(3):1–45.
18. Elasri M, et al. Image generation: A review. *Neural Process Lett*. 2022;**54**(5):4609–46.
19. Zhan F, et al. Multimodal image synthesis and editing: The generative AI era. *IEEE Trans Pattern Anal Mach Intell*. 2023;**45**(12):15098–119.
20. Beyan EVP, Rossy AGC. A review of AI image generator: influences, challenges, and future prospects for architectural field. *J Artif Intell Archit*. 2023;**2**(1):53–65.
21. Xie P, et al. Seismic inverse modeling method based on generative adversarial networks. *J Pet Sci Eng*. 2022;**215**:110652.
22. Gao Y, Liu X, Xiang J. FEM simulation-based generative adversarial networks to detect bearing faults. *IEEE Trans Ind Inf*. 2020;**16**(7):4961–71.
23. Qin S, et al. Intelligent design and optimization system for shear wall structures based on large language models and generative artificial intelligence. *J Build Eng*. 2024;**95**:109996.
24. Chen L, et al. How generative AI supports human in conceptual design. *Des Sci*. 2025;**11**:e9.
25. Li S, Farrar C, Yang Y. Efficient regional seismic risk assessment via deep generative learning of surrogate models. *Earthq Eng Struct Dyn*. 2023;**52**(11):3435–54.
26. Fui-Hoon Nah F, et al. Generative AI and ChatGPT: Applications, challenges, and AI-human collaboration. Taylor & Francis; 2023:277–304.
27. Kaveh A, et al. Chaotic swarming of particles: a new method for size optimization of truss structures. *Adv Eng Softw*. 2014;**67**:136–47.
28. Azegami H. *Shape Optimization Problems*. Springer; 2020.
29. Kaveh A. *Advances in Metaheuristic Algorithms for Optimal Design of Structures*. Springer; 2014.
30. Wu J, Sigmund O, Groen JP. Topology optimization of multi-scale structures: a review. *Struct Multidiscip Optim*. 2021;**63**:1455–80.
31. Yang RJ, Chen CJ. Stress-based topology optimization. *Struct Optim*. 1996;**12**(2):98–105.

32. Bendsoe MP, Sigmund O. *Topology Optimization: Theory, Methods, and Applications*. Springer Sci Bus Media; 2013.
33. Habibi A, Zarei S, Khaledy N. Optimum design of concrete gravity dams using sequential quadratic programming algorithm. *Dams Reservoirs*. 2021;**31**(1):10–20.
34. Aage N, Lazarov BS. Parallel framework for topology optimization using the method of moving asymptotes. *Struct Multidiscip Optim*. 2013;**47**(4):493–505.
35. Yu Z, et al. Structural optimization of truss using interior point and trust region. *Metal Int*. 2013;**18**(9):81.
36. Asadi E, et al. A TSK fuzzy model optimization with meta-heuristic algorithms for seismic response prediction of nonlinear steel moment-resisting frames. *Struct Eng Mech*. 2024;**90**(2):189–208.
37. Zhong R, et al. Surrogate ensemble-assisted hyper-heuristic algorithm for expensive optimization problems. *Int J Comput Intell Syst*. 2023;**16**(1):169.
38. Guo M, Hesthaven JS. Reduced order modeling for nonlinear structural analysis using Gaussian process regression. *Comput Methods Appl Mech Eng*. 2018;**341**:807–26.
39. Biancolini ME. *Fast Radial Basis Functions for Engineering Applications*. Springer; 2017.
40. Marasco S, Cimellaro GP. A new evolutionary polynomial regression technique to assess the fundamental periods of irregular buildings. *Earthq Eng Struct Dyn*. 2021;**50**(8):2195–211.
41. Asteris PG, et al. Prediction of the fundamental period of infilled RC frame structures using artificial neural networks. *Comput Intell Neurosci*. 2016;**2016**(1):5104907.
42. Fathnejat H, Ahmadi-Nedushan B. An efficient two-stage approach for structural damage detection using meta-heuristic algorithms and group method of data handling surrogate model. *Front Struct Civ Eng*. 2020;**14**:907–29.
43. Yang DY. Adaptive risk-based life-cycle management for large-scale structures using deep reinforcement learning and surrogate modeling. *J Eng Mech*. 2022;**148**(1):04021126.
44. Peter T. Using deep learning as a surrogate model in multi-objective evolutionary algorithms. Otto-von-Guericke-Universität Magdeburg; 2018.
45. Yu M, Li X, Liang J. A dynamic surrogate-assisted evolutionary algorithm framework for expensive structural optimization. *Struct Multidiscip Optim*. 2020;**61**(2):711–29.
46. Farhadi M, Ghiasi R, Torkzadeh P. Damage detection of truss structures using meta-heuristic algorithms and optimized group method of data handling surrogate model. *Struct*. 2024.
47. Delipetrev B, Tsinaraki C, Kostic U. Historical evolution of artificial intelligence. 2020.
48. Shaw J, et al. Artificial intelligence and the implementation challenge. *J Med Internet Res*. 2019;**21**(7):e13659.
49. Jordan MI, Mitchell TM. Machine learning: Trends, perspectives, and prospects. *Science*. 2015;**349**(6245):255–60.
50. Zhou ZH. *Machine Learning*. Springer Nature; 2021.
51. Yusuf S, et al. Fault classification improvement in industrial condition monitoring via Hidden Markov Models and Naïve Bayesian modeling. In: *2013 IEEE Symposium on Industrial Electronics & Applications*. IEEE; 2013.
52. Creswell A, et al. Generative adversarial networks: An overview. *IEEE Signal Process Mag*. 2018;**35**(1):53–65.

53. Girin L, et al. Dynamical variational autoencoders: A comprehensive review. *arXiv Preprint*. 2020. arXiv:2008.12595.
54. Naveed H, et al. A comprehensive overview of large language models. *ACM Trans Intell Syst Technol*. 2023.
55. Scopus. *Scopus: Abstract and Citation Database*. 2025 Jul 9.
56. Liao W, et al. Automated structural design of shear wall residential buildings using generative adversarial networks. *Autom Constr*. 2021;**132**:103931.
57. Lu X, et al. Intelligent structural design of shear wall residence using physics-enhanced generative adversarial networks. *Earthq Eng Struct Dyn*. 2022;**51**(7):1657–76.