buildings

MDPI

Article

A Physics-Informed Design Generator for Long-Span Reticulated Domes: Replacing Iterative Finite Element Analysis for Optimal Solutions

Xinyi Chen, Guozhi Qiu *, Jinghai Gong, Shanshan Shen and Yijie Zhang

School of Ocean and Civil Engineering, Shanghai Jiao Tong University, 800 Dongchuan Road, Minhang District, Shanghai 200240, China; nuyoah8@sjtu.edu.cn (X.C.)

* Correspondence: qiugz@sjtu.edu.cn; Tel.: +86-21-34207985

Abstract

The optimal design of long-span structures is hindered by the combination of prohibitively high computational costs and the limited physical consistency of purely data-driven surrogates. To address this challenge, this study proposes a multi-stage automated design framework that shifts the workflow from repeated per-task solving to reusable digital asset creation. First, a large-scale surrogate-optimized dataset containing 100,000 design samples is generated by embedding a high-speed MLP emulator into a Genetic Algorithm (GA). The core innovation lies in training a physics-regularized neural design generator. By incorporating a reduced-order total potential energy term derived from the principle of minimum potential energy as a regularization constraint, the network learns the mapping from external design conditions to validated near-optimal internal parameter combinations while suppressing mechanically unfavorable configurations associated with low stiffness. This mechanism improves mechanical admissibility, particularly in data-sparse regions. Validation results show that the generator achieves millisecond-level candidate generation and reduces the prediction error to 31% of that of conventional models under sparse-data conditions. In a like-for-like case study with identical external input parameters, the generated candidate design achieves a 21.1% reduction in total steel consumption. The proposed framework is therefore best understood as a rapid preliminary design tool for producing weight-efficient and mechanically admissible candidate schemes, which can then be subjected to subsequent high-fidelity analysis and code-based verification.

Keywords: three-centered reticulated dome; structural optimization; physics-informed machine learning; genetic algorithm; data-driven design; automated design

Check for updates

Academic Editor: Weixin Ren

Received: 5 March 2026
Revised: 31 March 2026
Accepted: 9 April 2026
Published: 23 April 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Long-span spatial structures form the structural basis of many large public buildings, such as stadiums, exhibition centers, and airport terminals. However, the design of these structures presents a formidable challenge. The optimization process must navigate a complex, high-dimensional design space where numerous variables are intricately coupled and exhibit non-linear relationships [1,2], posing significant risks to project timelines and budgets. This creates an urgent need for innovative automated design tools to ensure both the safety and economy of these critical assets.

The root of this challenge lies in the impracticality of an element-level approach, where directly optimizing thousands of nodal coordinates or member cross-sections becomes

intractable due to the curse of dimensionality. To overcome this fundamental barrier, this study proposes a paradigm shift in the problem formulation itself: elevating the optimization from the intractable element-level to a more efficient and holistic system-level through the establishment of a comprehensive parameter system. Taking the three-centered reticulated dome as a case, we use a finite set of key parameters to control the structure's macroscopic geometric form and microscopic features. This not only dramatically reduces the dimensionality of the design space but also lays the conceptual foundation for the efficient and reliable automated design framework that follows.

Another obstacle in this domain is the immense computational burden imposed by modern numerical methods. While Finite Element Analysis (FEA) is the gold standard for accurate simulation, its direct integration into an optimization loop—a paradigm often termed "FEA-in-the-loop"—is computationally prohibitive [3–5]. A single optimization, particularly when driven by robust global-search methods such as genetic algorithms, may require thousands of iterative simulations, leading to design cycles that are impractically long for real-world engineering projects.

To overcome this bottleneck, the research community has explored data-driven surrogate models, such as Multi-Layer Perceptron (MLP) and other deep neural networks, to approximate the FEA process and accelerate fitness evaluations [6–9]. Although these models offer high speed, they suffer from critical limitations. They are notoriously data-hungry, requiring massive volumes of FEA results for training. More critically, their black-box nature means their predictions may violate fundamental principles of structural mechanics, especially when extrapolating beyond the training data [10–13]. This lack of physical consistency compromises the trustworthiness of optimization results, posing potential risks to structural safety.

Physics-informed learning has recently emerged as a powerful paradigm for addressing these failings. By embedding physical principles directly into the training objective, such methods can improve generalization and reduce the risk of mechanically inconsistent predictions [14–16]. Among them, Physics-Informed Neural Networks (PINNs), in the classical sense, enforce physical consistency by incorporating governing-equation residuals into the loss function. However, the predominant research paradigm in structural optimization, often termed PINN-as-solver or PINN-TO, inadvertently creates a new computational bottleneck. Because the model must be retrained for each candidate design modification during optimization, this optimization-within-an-optimization loop can be extraordinarily time-consuming, rendering it impractical for rapid design exploration [17–21].

To realize the shift from the element-level to the system-level and to efficiently solve this new parameterized optimization problem, this study proposes an innovative, multi-stage framework to create a reusable, instantaneous design generator. Instead of merely using AI models to accelerate individual simulations, we strategically restructure the computational workflow to consolidate the prohibitive computational cost into a one-time, upfront investment, creating a reusable digital asset. Our methodology proceeds in three stages. First, we train a simple, high-speed MLP to act as an FEA emulator, whose sole task is to rapidly predict a key performance metric (total steel consumption). Second, we leverage this emulator within a GA-driven optimization to efficiently generate a large-scale surrogate-optimized design dataset of approximately 100,000 samples [22,23]. This unique dataset establishes the direct mapping from a vast array of external design conditions to their corresponding, pre-optimized internal design parameters.

The core innovation lies in the final stage: we train a physics-regularized design generator on this dataset of surrogate-optimized solutions. While these GA-optimized labels provide high-quality supervision, relying solely on data-driven imitation risks capturing numerical noise or spurious correlations. The role of physical regularization is therefore to

act as a mechanical filter. By embedding structural energy principles, the training process is elevated from simple statistical curve-fitting to physically guided learning. This ensures that the generator does not merely memorize discrete favorable design points but instead learns a mechanically admissible mapping that remains reliable in previously unexplored regions of the design space. In this study, the proposed model is not used as a classical PDE-residual PINN solver. Rather, it should be understood as a physics-regularized design generator within a broader physics-informed machine-learning framework, in which physical knowledge is introduced through an energy-based regularization term. Guided by a composite loss function that combines data-driven accuracy with a physics-based energy principle, the generator learns the complex nonlinear mapping from problem definition to validated near-optimal design recommendations. This approach consolidates the prohibitive computational cost into a one-time upfront investment to create a reusable digital asset—a trained physics-regularized design generator that can produce high-quality, mechanically admissible candidate designs in milliseconds for new tasks within the parameterized domain. It is important to emphasize that the present framework is intended for rapid preliminary design generation rather than for replacing the final code-based verification, member checking, and detailing process required in engineering practice. In this sense, the surrogate-in-the-loop strategy fundamentally decouples expensive offline training from rapid online candidate generation, thereby combining the reliability of physics-guided regularization with the speed of surrogate-based design generation.

The scientific merit and novelty of this research are encapsulated in the following three key contributions:

1. System-Level Problem Reformulation: We elevate the optimization problem from the intractable element-level to an efficient system-level through a comprehensive 18-parameter system, effectively overcoming the curse of dimensionality inherent in complex spatial structures.
2. Strategic Decoupling Architecture: Unlike existing PINN-as-solver frameworks that require expensive online retraining, our multi-stage workflow decouples global exploration (MLP-driven) from design-mapping learning (physics-regularized-generator-driven). This allows for the efficient creation of a large-scale surrogate-optimized dataset with validated near-optimal fidelity on representative sampled scenarios, focusing on the high-performance design landscape rather than mere response approximation.
3. Inductive Bias via Reduced-Order Energy Regularization: By embedding a reduced-order total potential energy functional derived from the principle of minimum potential energy into the loss function, the framework introduces a mechanics-based inductive bias without requiring online finite-element response recovery. This enables the generated designs to remain mechanically admissible and interpretable even in extrapolation regions where purely black-box models typically fail.

2. Literature Review

2.1. Optimization Methods for Complex Structures

The design of large-span complex structures is fundamentally an optimization problem, typically involving a search for optimal solutions within a high-dimensional, nonlinear, and often non-convex design space [1,2,24,25]. For such parametric optimization problems, where design spaces can be multi-modal and gradient information is difficult to obtain, heuristic algorithms inspired by natural phenomena offer a more robust path toward global optimization. Among heuristic global-search methods, genetic algorithms have been widely applied in structural optimization because of their robustness in handling non-convex and multimodal design spaces [5,22,23]. By simulating the evolutionary mech-

anism of "survival of the fittest," GA has demonstrated powerful global search capabilities, making it highly suitable for navigating the complex design landscapes characteristic of structures like reticulated domes.

However, despite the powerful search capabilities of algorithms like GA, a more fundamental computational bottleneck persists when applying them to practical structural optimization. Heuristic algorithms inevitably require a performance evaluation for every candidate design generated during the evolutionary process. In structural engineering, this evaluation typically necessitates a high-fidelity Finite Element Analysis (FEA) to accurately solve the governing Partial Differential Equations (PDEs) and assess the design's performance.

This requirement gives rise to the "FEA-in-the-loop" paradigm, where the computationally expensive FEA solver is directly embedded within the iterative optimization cycle. As the scale and complexity of the structure increase, the cost of obtaining a single high-precision PDE solution rises dramatically. Given that a single GA optimization run may require thousands or tens of thousands of such evaluations to adequately explore the design space, the entire process becomes extremely time-consuming, often lasting weeks or months. This prohibitive computational cost stands as the key obstacle to the widespread adoption of automated design in the AEC industry and severely limits its practical application in real-world projects. Consequently, to unlock the full potential of powerful optimization algorithms like GA, there is an urgent need for a new method that can fundamentally break the "FEA-in-the-loop" paradigm and enable efficient, near real-time performance evaluation.

2.2. Surrogate Modeling in Engineering Design

To address the computational bottleneck imposed by the "FEA-in-the-loop" paradigm, data-driven surrogate models have been extensively studied over the past two decades as an efficient alternative [5,26,27]. The core idea is to employ a computationally inexpensive approximate model, such as a Multi-Layer Perceptron (MLP) or other Deep Neural Networks (DNNs) [8,9,28,29], to learn and replace the complex input–output mapping of a high-fidelity simulation tool. Once trained, such surrogate models can provide near-instantaneous predictions, thereby significantly accelerating optimization.

However, despite their remarkable speed, these purely data-driven "black-box" surrogate models suffer from fundamental and unavoidable limitations, which severely restrict their reliable application in safety-critical structural engineering design. These limitations can be summarized in three key areas:

- Data-Hungry Nature: The accuracy of a surrogate model relies heavily on the scale and coverage of the training dataset. To achieve acceptable predictive performance across a broad design space, a very large number of samples must first be generated via high-fidelity simulations, which itself incurs a significant computational burden [30–33].
- Lack of Physical Consistency: Traditional surrogate models are purely data-driven black boxes. During training, they focus on minimizing output prediction errors without incorporating constraints from mechanics, such as equilibrium, compatibility, or energy conservation. As a result, especially in regions poorly covered by training data, they may produce physically inadmissible or unsafe designs [10,12,13,34].
- Poor Generalization and Robustness: The effectiveness of a traditional surrogate model is often limited to the interpolation range of the training samples, and their ability to generalize cannot be guaranteed [34–36]. When design conditions (e.g., load magnitudes, boundary positions) change, the model's performance may degrade rapidly, often necessitating costly data regeneration and model retraining, thereby exhibiting poor robustness and adaptability. Related studies have also explored transfer-learning

strategies to alleviate this retraining burden, although such approaches remain strongly task-dependent [33].

In summary, while conventional surrogate models represent a significant step forward in accelerating computation, their data-hungry and black-box nature prevents them from becoming a truly trustworthy tool for automated final design. They solve the problem of speed but introduce a new, critical problem of reliability. Therefore, the development of a new type of surrogate model that combines rapid prediction capability with inherent physical consistency has become the key to advancing the field of automated structural design. Recent studies have further explored a broad range of data-driven strategies for structural and topology optimization, including neural-network-based surrogate modeling, deep-learning-assisted topology generation, and data-efficient optimization workflows [37–48]. Collectively, these studies demonstrate the growing potential of learning-based methods in accelerating structural design and optimization. At the same time, they also highlight persistent challenges related to data dependence, transferability, and physical reliability, which further motivate the development of the present physics-regularized design generation framework.

2.3. Physics-Informed Machine Learning

To address the fundamental shortcomings of trustworthiness in traditional data-driven models, a key development in the field of Scientific Machine Learning (SciML)—Physics-Informed Machine Learning (PIML)—offers a novel pathway [16,49]. The Physics-Informed Neural Network (PINN), proposed by Raissi et al., stands as a representative technique. In its classical form, a PINN integrates the residual of the governing partial differential equations (PDEs) into the neural network loss function as a regularization term. Enabled by automatic differentiation, this mechanism compels the network output to adhere to physical laws across the entire domain while fitting observational data [15,50]. Consequently, it transforms a purely statistical regression problem into a more well-posed function approximation problem constrained by physical laws, significantly enhancing a model's generalization capability and the physical interpretability of its predictions [51–57].

In structural mechanics and design applications, however, physics-informed learning is not restricted to the classical PDE-residual formulation. Physical knowledge may also be incorporated through energy functionals, equilibrium relations, constitutive constraints, or other mechanically meaningful regularization terms. This broader perspective is particularly relevant when the objective is not to solve the governing field equations directly, but to learn a reliable design-oriented mapping under physical guidance.

In the realm of structural topology optimization, this powerful concept has inspired a new research paradigm, often referred to as "PINN-TO" or "PINN-as-solver" [20,21,58–67]. In this approach, the PINN is used to replace the traditional FEA solver entirely, constructing an end-to-end differentiable optimization framework. While this elegantly ensures physical fidelity, it inadvertently creates a new and severe computational bottleneck. Because the structural form or parameters change continuously during optimization iterations, the PINN must be completely retrained at every step to solve the PDE for each new candidate design. This "optimization-within-an-optimization" loop is an extremely time-consuming process. Numerical examples have shown that the time required for PINN-TO can be orders of magnitude greater than that of traditional methods, making it an impractical tool for efficient design exploration. The development of reusable computational libraries has also facilitated the broader implementation and dissemination of physics-informed learning methods in scientific computing [68].

Thus, a critical research gap becomes evident. The field is presented with two imperfect choices: fast but physically unreliable "black-box" surrogates, or physically consistent

but computationally prohibitive "PINN-as-solver" frameworks. This study is designed precisely to bridge this gap. Instead of using a neural network as an online PDE solver, we employ a pre-trained physics-regularized design generator whose physical prior is introduced through an energy-based regularization term. The model is trained on a large-scale surrogate-optimized dataset and is used for direct design generation under mechanical guidance rather than for field solution in the classical PINN sense.

This new "surrogate-in-the-loop" paradigm allows the physics-regularized design generator to learn the direct mapping from external design conditions to validated near-optimal internal parameter combinations. This approach simultaneously resolves both previously discussed dilemmas: it bypasses the "PINN-as-solver" bottleneck by consolidating the computation into a one-time offline training process, and it overcomes the reliability issues of black-box models by embedding physical principles into the final generator. Ultimately, it provides a novel solution for the automated design of complex structures—one that is efficient, reliable, and physically interpretable.

3. Methodology

To provide an overall view of the proposed workflow before detailing each component, the methodology is first summarized in Figure 1. As shown in the figure, the framework is organized into four sequential stages, including the construction of a high-speed FEA emulator, the generation of a surrogate-optimized design dataset, the training of a physics-regularized design generator, and the final application and validation through independent high-fidelity FEA. This staged workflow reflects the central idea of the present study, namely, shifting the major computational burden to an offline phase so that efficient and mechanically guided design generation can be achieved during subsequent use.

Stage 1: Building the High-Speed FEA Emulator. The framework begins by creating a foundational dataset through high-dimensional Sobol sampling across the entire 18-parameter design space, followed by high-fidelity FEA for each sample point. This dataset, mapping design parameters to steel consumption, is used to train a simple but extremely fast Multi-Layer Perceptron (MLP), which serves as an "FEA emulator" for performance prediction.

Stage 2: Generating the Surrogate-Optimized Design Dataset. Leveraging the millisecond-level prediction speed of the FEA emulator, a massive offline optimization campaign is conducted. A Genetic Algorithm (GA) is deployed to solve 100,000 independent design problems, each defined by a unique set of external conditions. For each task, the GA searches the internal-parameter space using the emulator as a fast fitness evaluator, thereby identifying high-quality surrogate-guided parameter combinations with greatly reduced computational cost. The result is a large-scale surrogate-optimized dataset that maps external design conditions directly to their corresponding internal design parameter combinations.

Stage 3: Training the physics-regularized design generator. This curated surrogate-optimized dataset is then used to train the core innovation of this framework—a physics-regularized neural design generator. Guided by a composite loss function that combines data-driven supervision with a structural energy principle, the model learns the direct mapping from design conditions to surrogate-optimized internal parameter combinations. In this way, the final generator is trained not merely to reproduce numerical labels, but to produce mechanically consistent and high-quality design recommendations.

Stage 4: Application and Validation. Once trained, the physics-regularized design generator can be deployed for near-instantaneous candidate generation on new design tasks. For any given set of 12 external parameters, it directly outputs a recommended set of surrogate-optimized internal parameters. The final step of the methodology involves a

rigorous validation protocol, in which the designs generated by the physics-regularized design generator are subjected to an independent high-fidelity FEA to verify their structural performance and engineering reliability.

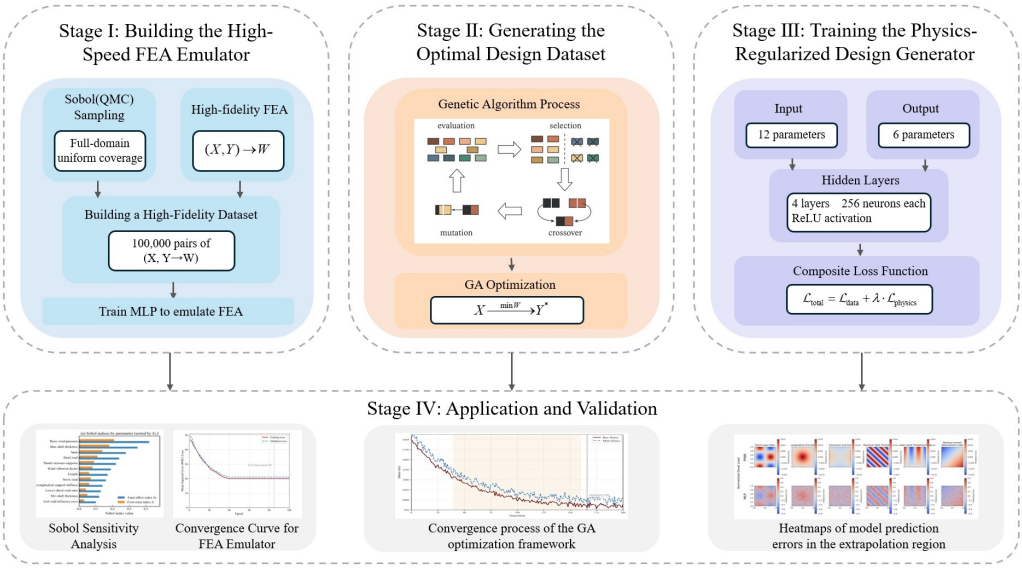
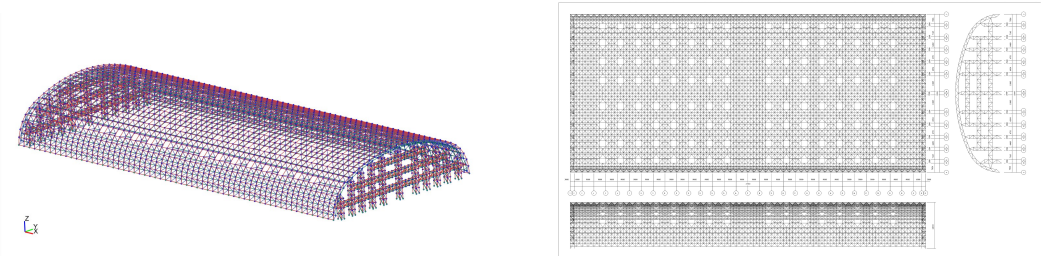


Figure 1. The integrated framework for automated design.

3.1. Parametric Modeling of Three-Centered Reticulated Domes

The subject of this study is the three-centered reticulated dome, a type of long-span spatial structure widely used in modern civil infrastructure. As illustrated in Figure 2, which presents both a comprehensive 3D Finite Element Model (FEM) and its corresponding 2D engineering drawings, the structure is characterized by a high degree of complexity, comprising many interconnected nodes and members forming a discrete bar-and-node system. This intrinsic complexity makes direct optimization at the element level computationally intractable and highlights the necessity for a more abstract and efficient modeling approach.



(**a**) (**b**)

Figure 2. Representative model of the three-centered reticulated dome structure. (**a**) 3D Finite Element Model (FEM). (**b**) Plan and elevation views.

3.1.1. Significance of Parametric Modeling

To effectively search for optimal solutions within the high-dimensional and complex design space of three-centered reticulated domes, this study adopts a parametric modeling approach. Directly optimizing the coordinates of hundreds or thousands of nodes or the cross-sectional dimensions of individual members would result in an exceptionally large and intractable number of design variables, a problem known as the "curse of dimensionality" [22,23]. Parametric modeling aims to capture the most essential geometric and structural characteristics of the dome using a finite and highly abstract set of parameters. This method elevates the optimization problem from the "element-level"

to the "system-level," which not only dramatically reduces the problem's dimensionality and computational complexity but also ensures that all generated design solutions are geometrically rational and vary continuously. This approach lays a solid foundation for the subsequent construction of the surrogate model and for efficient optimization.

3.1.2. Establishment of the Parameter System

Based on the design logic and engineering practices for reticulated dome structures, this study establishes a comprehensive system of 18 key parameters that cover multiple aspects, from macroscopic geometric form to microscopic structural features. The selection of these parameters adheres to four main principles: (1) Geometric Controllability: The parameters must clearly define the global and local form of the structure; (2) Mechanical Influence: The parameters must have a significant impact on the structure's mechanical performance, stiffness distribution, or stability; (3) Computational Feasibility: The parameters should be easy to assign and control within a numerical model; and (4) Engineering Practicality: The parameters must possess clear physical meaning, making them easy for designers to understand and apply.

Following these principles, we have categorized the 18 parameters into 12 external input parameters and 6 internal output parameters. The external parameters define the preconditions of the design problem (such as structural dimensions and loading conditions), while the internal parameters represent the optimal solution that the optimization algorithm is tasked with finding. The specific parameter system is detailed in Table 1.

Table 1. The comprehensive system of 18 key parameters for the parametric modeling of three-centered reticulated domes.

Category	Parameter	Symbol	Type	Design Role
Geometric form	Structural span	L	Input	Controls transverse extent of the grid-shell
	Structural length	B	Input	Controls longitudinal extent of the grid-shell
	Maximum crown elevation	$H_{\max}$	Input	Constrains the global height of the structure
Grid layout	Panels between supports	n_p	Input	Defines spacing of lower supports and overall mesh density
	Spanwise grid size (transverse)	g_s	Output	Governs member spacing/density along the span direction
	Longitudinal grid size	g_l	Output	Governs member spacing/density along the length direction
Structural features	Minimum shell thickness	$t_{\min}$	Input	Ensures stiffness at locally weak regions
	Maximum shell thickness	$t_{\max}$	Output	Ensures stiffness along the primary load-transfer paths
	Lower-chord thickening ratio	η_l	Output	Tunes local strengthening of the lower chord
	Permissible void ratio of lower chord	ϕ_l	Input	Controls stiffness distribution in discontinuous/voided zones
	Bending-moment redistribution index	γ_m	Output	Adjusts the ability to redistribute bending moments
Loading conditions	Dead load	q_d	Input	Represents permanent loading considered in design
	Snow load	q_s	Input	Represents roof live-load capacity due to snow
	Basic wind pressure	q_w	Input	Governs wind-load design basis
	Wind-vibration factor	ξ_w	Input	Accounts for dynamic effects of wind loading
Boundary & end-wall effects	Mesh rows within end-wall influence zone	N_e	Input	Controls extent of stiffness transfer from the end wall
	Longitudinal support stiffness	k_l	Input	Quantifies boundary restraint capacity along longitudinal edges
Derived design ratio	Rise-to-span ratio	λ_h	Output	Governs roof curvature and overall flatness

To provide a more intuitive understanding of the physical significance of these abstract parameters, Figure 3 offers a visual illustration of two representative examples from Table 1. The bending-moment redistribution index controls the effective structural depth near the dome's haunches. A lower value (e.g., $\gamma_m = 0.5$) implies a smaller effective depth, limiting the shell's capacity for moment redistribution and making it more bending-sensitive.

A higher value (e.g., $\gamma_m = 1.0$) increases the effective depth and enhances the structural stiffness and reserve capacity, enabling greater moment transfer from peak-stress regions to less-stressed sections (Figure 3a). The permissible void ratio of the lower chord defines the member density and material layout in the lower chord layer. The ratio specifies the proportion of nodes where members are intentionally removed (voided). Setting $\phi_l = 0$ maximizes redundancy and material use, while increasing ϕ_l (e.g., to 0.8) introduces systematic voids. This parameter is critical for the optimization algorithm to manage the trade-off between structural economy (material reduction) and system stiffness/redundancy (Figure 3b).

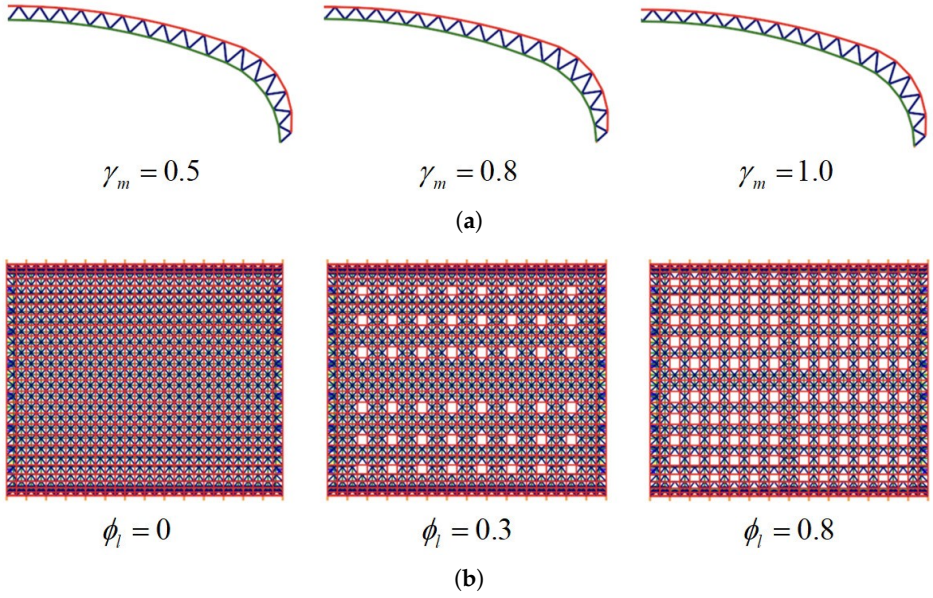


Figure 3. Visual illustration of representative design parameters and their physical effects. (**a**) Bending-moment redistribution index. (**b**) Permissible void ratio of lower chord.

3.1.3. Validation of the Parameter System's Effectiveness

To validate the effectiveness of the established parameter system and to gain a deeper understanding of each parameter's impact on the optimization objective, we conducted a Global Sensitivity Analysis (GSA). Considering the prevalent non-linear relationships and parameter coupling effects within the structural system, we employed the Sobol' method, which is based on variance decomposition. This method overcomes the limitations of traditional linear correlation analysis. By decomposing the total variance of the output variable (total steel consumption), the Sobol' method can quantify the direct contribution of each input parameter (the first-order Sobol' index, S_i) as well as the total contribution arising from that parameter's interactions with all other parameters (the total-effect index, S_{T_i}).

The analysis results (Figure 4) indicate that all 18 parameters have varying degrees of influence on the total steel consumption. Parameters related to geometric form (such as the rise-to-span ratios) and grid dimensions exhibit high values for both their first-order and total-effect indices, confirming them as key drivers influencing the structural economy. More importantly, the total-effect indices (S_{T_i}) for multiple parameters are significantly higher than their first-order indices (S_i), which clearly reveals the presence of strong interactions among the parameters. For example, the influence of grid size on steel consumption varies with changes in the structural span. This finding strongly validates that our chosen parameter set is not only relevant but also comprehensive, capable of capturing the system's complex internal coupling mechanisms. It thereby provides a robust, reliable, and information-rich foundation for the subsequent surrogate modeling and optimization processes.

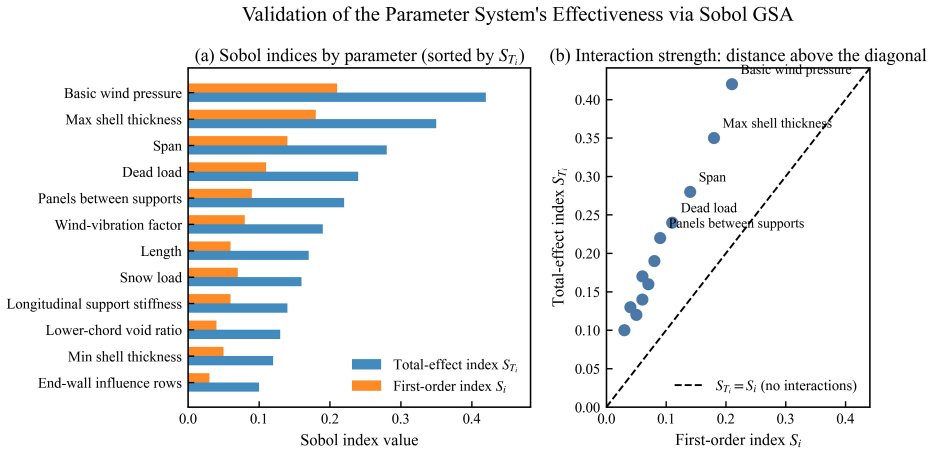


Figure 4. Validation of the parameter system's effectiveness via Sobol Global Sensitivity Analysis (GSA).

3.2. Building the High-Speed Performance Predictor (FEA Emulator)

3.2.1. High-Dimensional Sampling and High-Fidelity Data Acquisition

The foundational prerequisite for training a reliable FEA Emulator is the creation of a comprehensive, high-fidelity dataset that accurately represents the behavior of the structural system across its entire design space. To this end, a systematic data generation strategy was employed, combining a high-dimensional, quasi-random sampling method with a large number of automated Finite Element Analysis (FEA) simulations.

The process began with defining the design space, which, as established in Section 3.1, consists of a total of 18 key parameters (12 external and 6 internal). To ensure a thorough and uniform exploration of this high-dimensional space, the Sobol sequence—a Quasi-Monte Carlo (QMC) method—was selected for sampling. Unlike traditional pseudo-random sampling, which can lead to clustering and sparse regions, the low-discrepancy nature of the Sobol sequence guarantees a more homogeneous coverage of the parameter space. A total of 100,000 unique sample points, each representing a complete 18-dimensional design vector, were generated within the predefined valid range. The selection of this sample size (100,000) is governed by the need to ensure sufficient coverage across the 18-dimensional design space. According to the properties of the Sobol sequence in high-dimensional domains, this threshold was determined to reach a target point density that minimizes the discrepancy of the sampling distribution, thereby ensuring that the subsequent FEA Emulator can capture the complex, non-linear mechanical boundaries without significant 'blind spots.' While computationally intensive, this volume of data provides the high-fidelity foundation required to build a robust digital asset capable of handling extreme or rare project scenarios. To ensure the practical relevance of the dataset, the design domain was defined based on representative engineering specifications for long-span domes: structural spans range from 80 m to 150 m, structural lengths from 150 m to 300 m, and total steel consumption W typically fluctuates between 600 t and 1500 t depending on loading conditions. to ensure the transparency and reproducibility of the data generation process.

Subsequently, for each of the 100,000 parameter combinations, a high-fidelity data acquisition process was executed. This involved an automated workflow where each 18-dimensional vector was used to parametrically generate a complete structural model. A full Finite Element Analysis was then performed on this model to precisely calculate its corresponding key performance indicator: the total structural steel consumption. This computationally intensive process resulted in a final dataset containing 100,000 high-quality samples. Each sample consists of an 18-dimensional input vector (the design parameters)

and its corresponding 1-dimensional scalar output (the total steel consumption), forming the basis for the supervised learning task in the subsequent section.

3.2.2. Emulator Architecture and Training (MLP)

Following the creation of the high-fidelity dataset, the next step is to train a surrogate model to serve as the high-speed FEA Emulator. A Multi-Layer Perceptron (MLP), a type of feed-forward neural network, was chosen for this task due to its proven effectiveness as a universal function approximator. The entire process, from data preparation to model training and saving, was implemented using the TensorFlow and Keras libraries.

Data Preprocessing: Before training, the dataset was systematically prepared. It was first partitioned into a training set and a testing set, with 80% of the 100,000 samples allocated for training and the remaining 20% reserved for final model evaluation. To ensure stable and efficient convergence during training, the 18-dimensional input features were normalized using the StandardScaler from the Scikit-learn library. The scaler, fitted only on the training data to prevent data leakage, was saved alongside the final model to process new data during subsequent stages.

Emulator Architecture and Training: The architecture of the MLP was determined through a systematic trial-and-error process guided by the performance on the validation set. We evaluated various configurations (varying depth from 2 to 6 layers and width from 64 to 512 neurons) and monitored the convergence behavior. The final structure—three hidden layers with 128, 256, and 128 neurons, respectively—was selected as it represents the 'elbow point' where further increases in model complexity yielded negligible improvements in Mean Squared Error (MSE) while increasing the risk of overfitting. The Rectified Linear Unit (ReLU) was selected as the activation function to provide a robust non-linear gradient flow, while the Adam optimizer with an initial learning rate of 0.001 ensured a balanced convergence speed.

3.3. Design Dataset Driven by Genetic Algorithm

3.3.1. Algorithm Principle and Applicability

With the high-speed FEA Emulator established in Stage 1, the framework proceeds to its second critical phase: the large-scale generation of a surrogate-optimized design dataset. The primary objective of this stage is not to claim exact finite-element optima for all design scenarios, but to systematically map a broad landscape of external design conditions to their corresponding emulator-guided high-quality internal design parameters. To achieve this, 100,000 independent optimization tasks are formulated, each defined by a unique set of 12 external parameters generated via Sobol sampling. For each task, a Genetic Algorithm (GA), guided by the instantaneous predictions of the FEA Emulator, is employed to efficiently search the 6-dimensional space of internal parameters and identify the parameter combination that minimizes the emulator-predicted total steel consumption. The final output of this stage is a comprehensive dataset containing approximately 100,000 pairs of input external conditions and surrogate-optimized internal design parameters, which serves as the training data for the final Design Generator in the next stage.

For the optimization problem of the three-centered reticulated dome structure in this study, the applicability of the Genetic Algorithm is primarily manifested in the following aspects:

1. Global Search Capability in High-Dimensional Complex Solution Spaces: The optimization problem in this research involves 12 external design parameters, constituting a high-dimensional design space. The introduction of the surrogate-guided optimization framework further reveals the complex non-linear mapping relationships between these parameters and the structural performance. GA maintains a diverse

population, conducting parallel searches across multiple regions of the solution space in each iteration. Its crossover operator effectively integrates the superior traits of parent individuals (exploitation), while the mutation operator introduces new genetic information into the population, exploring untouched regions of the solution space (exploration). This mechanism, which combines exploration and exploitation, endows GA with an exceptional global search capability, enabling it to effectively avoid the trap of local optima caused by the non-convexity of the objective function, thereby significantly increasing the probability of finding the globally optimal design solution.

2. Applicability to Gradient-Free Objective Functions: The fitness function in this study (i.e., the structural steel consumption) is evaluated through a pre-trained FEA Emulator. This surrogate model is essentially a complex "black box" that cannot provide an analytical gradient of its output with respect to its input. As a typical "black-box" optimizer, the evolutionary process of a Genetic Algorithm is entirely independent of the objective function's gradient information; instead, it guides the search direction solely based on the fitness value corresponding to each individual. This characteristic allows the Genetic Algorithm to be seamlessly integrated with the pre-trained FEA Emulator, transforming the emulator's rapid prediction capability into the driving force for efficient GA-based search, which aligns well with the optimization framework adopted in this study.
3. Inherent Robustness and Flexibility: The search process of a Genetic Algorithm has an inherent randomness, making it insensitive to the selection of the initial population and demonstrating strong robustness. At the same time, its algorithmic framework is highly flexible and scalable. Key parameters such as encoding methods, population size, and genetic operator probabilities can be easily adjusted according to the specific problem to balance the algorithm's convergence speed and solution accuracy. For complex engineering problems like structural optimization, this robust and flexible nature makes GA a reliable and efficient engine for data generation. Crucially, the proposed multi-stage framework is designed to be 'optimizer-agnostic.' While this study employs the Genetic Algorithm as a representative tool due to its established efficacy in spatial structures, the GA functions merely as a modular 'sampling engine' within the Stage 2 campaign. The core innovation resides in the decoupled workflow architecture, which can seamlessly accommodate modern metaheuristics—such as LSHADE, CMA-ES, or Adaptive Particle Swarm Optimization—depending on the specific requirements of the design space complexity.

The applicability of the Genetic Algorithm is particularly pronounced in this stage, where the goal is to solve a massive number of independent optimization tasks in parallel. The "black-box" nature of GA, which relies only on fitness values rather than gradient information, allows it to be seamlessly integrated with the FEA Emulator from Stage 1. Its powerful global search capability, which balances exploration and exploitation, is crucial for efficiently identifying high-quality solutions across the entire spectrum of the 100,000 predefined design scenarios. This makes GA the ideal engine to power the automated data generation workflow, transforming the rapid predictive power of the emulator into a robust mechanism for creating the Optimal Design Dataset.

3.3.2. Integration of the FEA Emulator and GA for Fitness Evaluation

The core strategy for enabling the large-scale data-generation task is the integration of the pre-trained FEA Emulator from Stage 1 into the GA workflow as a fast fitness evaluator. In each of the 100,000 optimization tasks, an individual chromosome is encoded as a 6-dimensional real-valued vector corresponding to the internal design parameters. During the evolutionary search, instead of invoking the computationally expensive direct

finite-element solver for every candidate solution, the GA queries the trained FEA Emulator to obtain an instantaneous prediction of structural steel consumption.

Accordingly, the fitness function used in Stage 2 can be written as

$$W = f_{\text{FEA-Emulator}}(x_{\text{fixed}}, y), \tag{1}$$

where y denotes the set of six internal design parameters represented by the current chromosome, x_{fixed} denotes the corresponding set of externally specified design conditions for the current optimization task, $f_{\text{FEA-Emulator}}(\cdot)$ represents the trained surrogate model established in Stage 1, and W is the predicted total steel weight returned by the emulator. The GA then minimizes this predicted objective value through iterative selection, crossover, and mutation, thereby searching for high-quality solutions on the emulator-defined fitness landscape.

Under this formulation, the resulting labels generated in Stage 2 should be interpreted as surrogate-optimized design samples rather than exact finite-element optima. In other words, the GA identifies internal parameter combinations that are optimal with respect to the surrogate model prediction, while the fidelity of these labels with respect to direct finite-element re-optimization is further examined separately in the validation stage described later in the manuscript. This substitution replaces a multi-hour FEA call with a millisecond-level prediction, making the execution of 100,000 optimization runs computationally feasible and enabling the construction of a large-scale surrogate-optimized design dataset.

3.3.3. Algorithm Parameter Configuration

The performance of a Genetic Algorithm is largely dependent on the configuration of its key control parameters. For the massive optimization task in this study, a carefully calibrated combination of parameters is crucial to ensuring that the algorithm can converge efficiently and stably for each independent run. The settings, detailed in Table 2, were uniformly applied to each of the 100,000 optimization tasks to ensure consistency throughout the dataset generation process. The specific settings are configured as shown in Table 2.

Table 2. Algorithm Parameter Configuration.

Parameter	Value	Description
Population size	150	Number of individuals per generation.
Maximum generations	300	Termination criterion.
Crossover probability	0.90	Probability of applying crossover to a mating pair.
Mutation probability	0.02	Probability of applying mutation to offspring (real-coded genes).
Encoding scheme	Real-coded	Continuous chromosomes for design variables.
Selection strategy	Elitism	Retains the best individual(s) each generation.

The rationale for the selection of these parameters is as follows:

The selection of the GA parameters follows established heuristic principles for high-dimensional structural optimization and was further refined through preliminary sensitivity calibrations. The population size was set to 150—roughly 12.5 times the number of design variables—to ensure a sufficiently diverse gene pool while maintaining computational efficiency. Preliminary pilot runs indicated that a crossover probability of 0.9 and a mutation probability of 0.02 provided the optimal balance between exploration of the design space and exploitation of high-quality solutions. If the problem's dimensionality were to increase

significantly, the population size and max generations would be scaled proportionally according to the $O(n^2)$ complexity of the search space to maintain global search stability.

3.4. Training the Final Physics-Regularized Design Generator

This stage represents the core innovation of the proposed framework: the construction of a physics-regularized neural design generator that functions not as a performance predictor, but as an instantaneous design generator. The fundamental goal is to train a model that learns the direct, non-linear mapping from a given set of 12 external design conditions to the corresponding 6 surrogate-optimized internal design parameters.

The training process is guided by a novel composite loss function. This composite loss function is designed to synergize empirical evidence with axiomatic mechanical knowledge. The integration of the physics-based term is not redundant; rather, it provides a crucial inductive bias that governs the learning manifold. Even when the training labels are pre-optimized, the physics loss acts as a structural regularizer that guides the network toward inherently stiffer and more stable configurations. In essence, it serves as a physically meaningful regularization mechanism that fills the information gaps between sparse GA samples and helps ensure that the generated outputs remain mechanically plausible and self-consistent. Crucially, the physics loss is not derived from a specific PDE, but from a more fundamental structural energy principle. By transforming the training into a constrained function approximation problem, this approach improves generalization and physical consistency, enabling the final design generator to produce high-quality preliminary design recommendations with near-instantaneous inference.

3.4.1. Mathematical Formulation of the Design Generator

(1) Neural Network Architecture: This study employs a fully connected feed-forward neural network (MLP) to approximate the mapping function from the 12 external input parameters to the 6 surrogate-optimized internal design parameters. The specific architecture is as follows:

- Input Layer: 12 neurons, receiving the normalized external design parameters x.
- Hidden Layers: 4 hidden layers, each with 256 neurons, using the Rectified Linear Unit (ReLU) as the activation function.
- Output Layer: 6 neurons with a linear activation function, directly outputting the predicted surrogate-optimized internal parameters $\hat{y}$.

(2) Composite Loss Function: The training of the physics-regularized design generator is governed by a composite loss function, $\mathcal{L}_{\text{total}}$, which orchestrates a critical trade-off between fitting the surrogate-optimized labels for weight minimization and adhering to structural mechanics laws through stability-oriented regularization. It is defined as:

$$\mathcal{L}_{\text{total}}(\boldsymbol{\theta}) = \mathcal{L}_{\text{data}}(\boldsymbol{\theta}) + \lambda \cdot \mathcal{L}_{\text{physics}}(\boldsymbol{\theta}) \tag{2}$$

where $\boldsymbol{\theta}$ denotes the trainable parameters of the neural network and λ is a dynamic weighting coefficient.

- Data Loss Term ($\mathcal{L}_{\text{data}}$): This term acts as the primary driver for optimality. It compels the network to approximate the specific mapping from external conditions to the minimal-weight design parameters found by the GA. It is computed as the Mean Squared Error (MSE):

$$\mathcal{L}_{\text{data}} = \frac{1}{N}\sum_{i=1}^{N}\|\hat{\mathbf{y}}_i(\mathbf{x}_i) - \mathbf{y}_i^*\|^2 \tag{3}$$

where N is the batch size, $\mathbf{x}_i$ is the input vector of external conditions, $\hat{\mathbf{y}}_i$ is the predicted internal parameter vector, and $\mathbf{y}_i^*$ is the reference surrogate-optimized vector from the dataset.

- Physics-Based Regularization Term (L_{physics}): To ensure that the generated designs remain structurally admissible, particularly in regions where training data are sparse or noise-prone, a physics-based regularization term derived from the principle of minimum potential energy is introduced. Rather than expressing the strain energy through member-level internal forces, which would require explicit recovery of axial forces and bending moments for all structural members, the present study adopts a reduced-order total potential energy representation constructed directly from the parameterized description of the three-centered reticulated dome. In this way, the physical prior is introduced through an energy-based consistency principle rather than a PDE-residual formulation [69].

  For each training sample, the admissible displacement field is approximated by a small set of generalized coordinates,

$$\mathbf{q} = [q_c,\ q_m,\ q_e]^{\mathrm{T}}, \tag{4}$$

  associated with the dominant global deformation characteristics of the dome, namely the crown-dominated global bending mode, the meridional–ring coupling mode, and the end-wall/support interaction mode. The physics loss is then defined from the corresponding reduced compliance:

$$L_{\text{physics}} = \frac{1}{N}\sum_{i=1}^{N} C_r(\mathbf{x}_i, \hat{\mathbf{y}}_i), \qquad C_r(\mathbf{x}_i, \hat{\mathbf{y}}_i) = \frac{1}{2}\mathbf{f}_r(\mathbf{x}_i)^{\mathrm{T}}\mathbf{K}_r(\mathbf{x}_i, \hat{\mathbf{y}}_i)^{-1}\mathbf{f}_r(\mathbf{x}_i). \tag{5}$$

  This expression follows from the reduced total potential energy

$$\Pi_r(\mathbf{q}; \mathbf{x}_i, \hat{\mathbf{y}}_i) = \frac{1}{2}\mathbf{q}^{\mathrm{T}}\mathbf{K}_r(\mathbf{x}_i, \hat{\mathbf{y}}_i)\mathbf{q} - \mathbf{f}_r(\mathbf{x}_i)^{\mathrm{T}}\mathbf{q}, \tag{6}$$

  whose stationarity condition,

$$\frac{\partial \Pi_r}{\partial \mathbf{q}} = 0, \tag{7}$$

  yields

$$\mathbf{q}^* = \mathbf{K}_r(\mathbf{x}_i, \hat{\mathbf{y}}_i)^{-1}\mathbf{f}_r(\mathbf{x}_i). \tag{8}$$

  Therefore, the regularization remains fully consistent with the minimum-potential-energy principle, while avoiding member-by-member response recovery during training. The reduced stiffness matrix is assembled as

$$\mathbf{K}_r = \sum_{z\in\mathcal{Z}} (\mathbf{K}_{m,z} + \mathbf{K}_{b,z}) + \mathbf{K}_{\text{sup}}, \tag{9}$$

  where $\mathcal{Z}$ denotes the principal stiffness zones implied by the parameterized dome configuration, including the field zone, the haunch/moment-redistribution zone, the lower-chord enhancement zone, and the end-wall influence zone. For each zone z, the generalized membrane and bending stiffness contributions are written as

$$(\mathbf{K}_{m,z})_{ab} = EA_z\alpha_{z,ab}, \qquad (\mathbf{K}_{b,z})_{ab} = EI_z\beta_{z,ab}. \tag{10}$$

Here, A_z and I_z are the equivalent membrane and bending section measures of zone z. They are determined explicitly from the known parameter system through

$$A_z = \rho_z t_z, \qquad I_z = \rho_z \frac{t_z^3}{12}, \tag{11}$$

in which t_z is the equivalent thickness of the corresponding zone and ρ_z is the projected stiffness density implied by the grid spacing and topology continuity. In the present parameterization,

$$t_f = t_{\min}, \qquad t_c = t_{\max}, \qquad t_h = t_{\min} + \gamma_m(t_{\max} - t_{\min}), \qquad t_l = t_{\min}[1 + \eta_l(1 - \phi_l)], \tag{12}$$

for the field zone, crown-sensitive zone, haunch redistribution zone, and lower-chord enhancement zone, respectively. The projected stiffness density ρ_z is determined directly from g_s, g_l, and ϕ_l, while the width of the end-wall influence zone is determined by $N_e g_l$. The boundary restraint contribution is represented by

$$\mathbf{K}_{\text{sup}} = k_l \mathbf{T}_{\text{sup}}, \tag{13}$$

where $\mathbf{T}_{\text{sup}}$ is the support projection matrix associated with the admissible displacement basis.

The coefficients $\alpha_{z,ab}$ and $\beta_{z,ab}$ are not empirical fitting constants. They are generalized stiffness coefficients obtained from the Ritz energy formulation, namely from the domain integrals of the selected admissible shape functions and their derivatives over each zone:

$$\alpha_{z,ab} = \int_{\Omega_z} B_{m,a}^T B_{m,b} \, \mathrm{d}\Omega, \qquad \beta_{z,ab} = \int_{\Omega_z} B_{b,a}^T B_{b,b} \, \mathrm{d}\Omega. \tag{14}$$

Once the admissible displacement basis and the zone partition are specified, these coefficients become deterministic reduced-order quantities and can be evaluated offline before network training. In the same manner, the support projection matrix T_{sup} is prescribed by the support configuration, while the equivalent thickness t_z and the projected stiffness density ρ_z are obtained directly from the parameterized geometric and topological variables through Equations (10)–(12). The reduced load vector $f_r(x)$ is assembled from the prescribed external actions represented by the input variables, including the gravity- and wind-related design conditions, and is projected onto the same admissible basis. Consequently, for each training sample, both $K_r(x, \hat{y})$ and $f_r(x)$ are constructed directly from the known input x and the network-predicted internal parameters $\hat{y}$ through the reduced-order formulation itself. No member-level force recovery, no additional finite-element analysis, and no calls to the Stage 1 MLP-based FEA emulator are involved during design-generator training.

- Physical Interpretation and Mechanism: Within this formulation, a smaller reduced compliance corresponds to a smaller minimum total potential energy and thus to a stiffer and mechanically more admissible structural state under the prescribed loading. The role of L_{physics} is therefore not to replace the weight-minimizing tendency learned from L_{data}, but to regularize the learned mapping against low-stiffness parameter combinations that may arise from purely statistical interpolation or extrapolation. This regularization is particularly important in sparsely sampled regions, where a data-driven model may otherwise produce geometrically plausible yet mechanically unfavorable solutions. By penalizing the reduced compliance, the network is guided toward parameter combinations that preserve efficient load transfer and adequate global stiffness while remaining consistent with the supervised optimal design labels.

- Balancing Factor (λ): A critical challenge in training the physics-regularized design generator is the multi-scale nature of the loss landscape, where the gradients of the data loss $\mathcal{L}_{data}$ and the physics loss $\mathcal{L}_{physics}$ can differ by several orders of magnitude [16,55,70]. To ensure stable convergence, we employ a Gradient-based Dynamic Weighting strategy inspired by adaptive training methods in physics-informed learning [70]. The balancing factor λ is updated every k iterations to ensure that the contribution of the physics-based regularization is adaptively aligned with the supervised data-driven signal. The update rule is formulated as follows:

$$\lambda^{(t+1)} = \alpha\lambda^{(t)} + (1-\alpha)\frac{\overline{\|\nabla_\theta \mathcal{L}_{data}\|}}{\overline{\|\nabla_\theta \mathcal{L}_{physics}\|}}, \tag{15}$$

  where $\nabla_\theta \mathcal{L}$ denotes the gradient of the loss with respect to the network parameters θ, and $\|\cdot\|$ represents the Frobenius norm. $\overline{\|\cdot\|}$ signifies the moving average of the gradient magnitudes over a look-back window (e.g., $k = 100$ epochs) to prevent stochastic oscillations. The hyperparameter α is a momentum factor (set to 0.9 in this study) that ensures a smooth transition of the weight. By dynamically equalizing the gradient magnitudes, this mechanism prevents the optimizer from being biased toward the empirical data in the early stages or being trapped in physically consistent but suboptimal local minima, thereby fostering a robust mapping toward the optimality manifold [70].

3.4.2. Model Training and Implementation

To ensure that the training process of the physics-regularized design generator is both efficient and reproducible, we formulated a detailed implementation strategy. The entire model is implemented using the TensorFlow 2.x and Keras libraries. All training and evaluation procedures are conducted on a workstation equipped with NVIDIA RTX series GPUs to leverage their parallel computing capabilities and accelerate the training process. During this stage, the optimizer evaluates only the composite loss of the design generator. All quantities entering $L_{physics}$ are either pre-evaluated reduced-order coefficients or algebraic functions of the external input x and the predicted internal parameters $\hat{y}$. Therefore, the training loop does not contain online finite-element solves or surrogate-emulator queries.

- Optimizer Strategy: Considering the complexity of the physics-regularized loss landscape, this study employs a hybrid optimizer strategy. In the initial phase of training (approximately the first 80% of epochs), the Adam optimizer is used to leverage its adaptive learning rate for rapid convergence, quickly guiding the model toward the region of the optimal solution. In the later stages of training, the optimizer is switched to L-BFGS, a quasi-Newton method that utilizes second-order information for more refined local searching, thereby further reducing the loss value and improving convergence accuracy.
- Training Hyperparameter Settings:
  - Epochs: The model is trained for a total of 5000 epochs.
  - Learning Rate: The initial learning rate for the Adam optimizer is set to 0.001, with an exponential decay strategy applied. The learning rate is multiplied by 0.9 every 1000 epochs to ensure stability in the later stages of training.
  - Weight Initialization: The network weights are initialized using the Xavier initialization method. This method adjusts the initial variance of the weights based on the number of input and output neurons, aiming to maintain a stable signal

magnitude as it propagates through the network, which effectively prevents the problems of vanishing or exploding gradients.

- Data Processing and Splitting: The original dataset is randomly split into a training set and a validation set at an 8:2 ratio. The training set is used to update the network parameters, while the validation set, which does not participate in gradient updates during training, is used solely to monitor the model's generalization performance, help identify potential overfitting and provide an objective basis for hyperparameter selection.

4. Validation and Results

4.1. Performance Evaluation Metrics

To quantitatively assess the predictive performance of the surrogate models developed in Stage 1 (FEA Emulator) and Stage 3 (Design Generator), we employed three standard evaluation metrics widely used in regression tasks: Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and the Coefficient of Determination (R^2). Their mathematical definitions are as follows:

1. Mean Absolute Error (MAE):

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{16}$$

2. Root Mean Squared Error (RMSE):

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{17}$$

3. Coefficient of Determination (R^2):

$$R^2 = 1 - \frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y})^2} \tag{18}$$

where $\hat{y}_i$ and y_i are the predicted and true values, respectively, $\bar{y}$ is the mean of the true values, and n is the number of test samples.

To comprehensively evaluate the performance advantages of the physics-regularized design generator developed in this study for the design optimization of three-centered reticulated domes, this paper conducts systematic comparative experiments against a traditional Multi-Layer Perceptron (MLP) model. The analysis is carried out from three perspectives: prediction accuracy on the full dataset, data efficiency under varying sample sizes, and generalization capability in unseen design regions. Through quantitative metrics and visual comparisons, we validate that the proposed generator, while maintaining high accuracy, possesses superior data adaptability and stronger physical consistency than the purely data-driven baseline.

4.2. Validation of the High-Speed FEA Emulator

This section presents the experimental results for Stage 1 of the framework, focusing on the performance validation of the Multi-Layer Perceptron (MLP) model trained to function as the high-speed FEA Emulator. The objective is to demonstrate that this emulator can accurately and reliably predict the total steel consumption from the 18 input parameters, thereby confirming its suitability for the large-scale data generation task in Stage 2.

4.2.1. Training Process and Convergence

The MLP emulator was trained using the procedure detailed in Section 3.2.2. The training process was monitored by observing the Mean Squared Error (MSE) loss on both the training data and a separate validation set. Figure 5 illustrates the convergence curves over the training epochs.

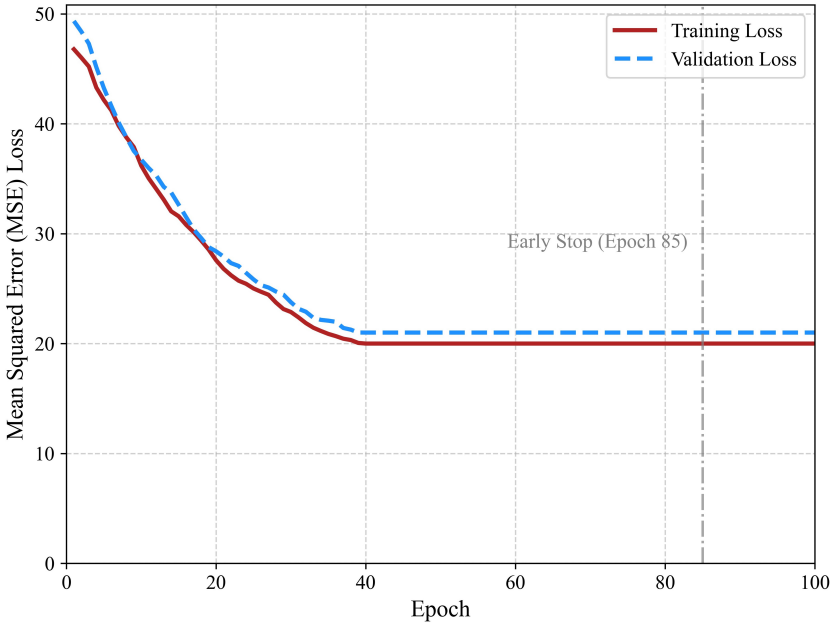


Figure 5. Training and Convergence Curve for FEA Emulator.

As shown in the figure, both the training and validation losses decrease rapidly during the initial epochs, indicating that the model was effectively learning the underlying patterns in the data. After this initial phase, the curves begin to plateau, and the validation loss remains stable without any significant increase, suggesting that no overfitting has occurred. The training was automatically concluded by the Early Stopping callback when the validation loss ceased to improve, ensuring that the model with the best generalization performance was retained. This stable convergence behavior confirms that the selected network architecture and training strategy were appropriate for the task.

4.2.2. Performance Evaluation on the Test Set

Following the successful training, the final performance of the FEA Emulator was quantitatively assessed on the unseen test set, which comprised 20,000 unique data samples. The evaluation was based on the three standard metrics defined in Section 4.1: MSE, MAE, R^2. The results are summarized in Table 3.

Table 3. Model Performance Metrics.

Metric	Value
Mean Squared Error (MSE)	23.325
Mean Absolute Error (MAE)	3.553
Coefficient of Determination (R^2)	0.98

These results indicate that the trained MLP model is sufficiently accurate to serve as an efficient surrogate evaluator in the Stage 2 GA-driven search. At the same time, because the optimization in Stage 2 is still performed on emulator predictions rather than direct finite-element evaluations, the resulting dataset should be interpreted as a surrogate-optimized

dataset. Residual surrogate error may therefore propagate into the optimization labels, and the fidelity of these labels with respect to direct finite-element re-optimization is further examined in the subsequent validation analysis.

4.3. Analysis of the Optimal Design Dataset Generation

4.3.1. Convergence Analysis of the GA-Driven Search

To evaluate the stability of the Stage 2 search process, the convergence behavior of the GA was analyzed on the emulator-defined fitness landscape. Figure 6 presents the convergence curves of the best fitness and the population-average fitness for a representative optimization run. In the early generations, both curves decrease rapidly, indicating efficient exploration of the design space and effective elimination of inferior candidate solutions. In the later generations, the best-fitness curve gradually plateaus and the gap between the best and average fitness narrows, indicating convergence toward a stable high-performance region of the surrogate-defined solution space.

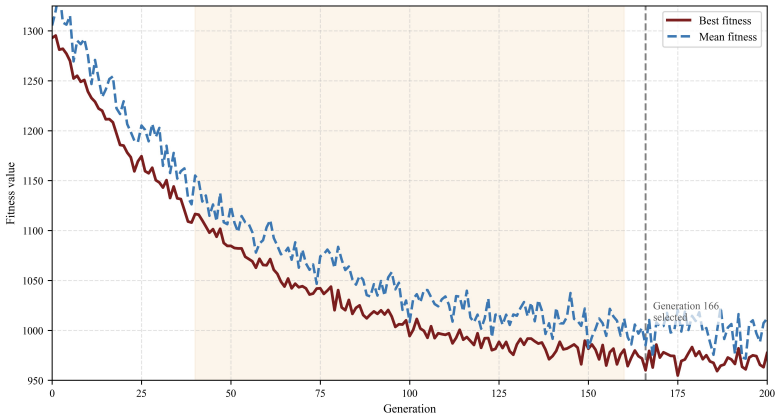


Figure 6. Convergence process of the GA optimization framework.

These results show that the GA settings used in Stage 2 were sufficient to obtain stable emulator-guided solutions for the sampled design scenarios. However, convergence on the surrogate-defined fitness landscape alone does not establish the fidelity of the resulting labels with respect to direct finite-element optimization. For this reason, an additional direct FEA validation was performed, as presented in the following subsection.

4.3.2. Direct FEA Validation of the Fidelity of the Surrogate-Optimized Dataset

To assess the fidelity of the Stage 2 labels with respect to direct finite-element optimization, an independent validation was conducted on a representative subset of the surrogate-optimized dataset. Specifically, 100 design scenarios were selected from the 100,000 Stage 2 samples so as to preserve a broad coverage of the Sobol-sampled external-condition space. For each selected scenario, two solutions were compared. The first solution was the surrogate-guided optimum generated in Stage 2, denoted as $y^{(s)}$. This solution was re-evaluated using direct finite element analysis to obtain its true structural objective value, $W_{\mathrm{FEA}}(y^{(s)})$. The second solution, denoted as $y^{(f)}$, was obtained by rerunning the GA under the same external conditions, but with direct FEA used as the fitness evaluator throughout the optimization process. This second solution was treated as the reference optimum for validation, with objective value $W_{\mathrm{FEA}}(y^{(f)})$.

The fidelity of the surrogate-optimized labels was quantified using the optimality gap

$$\mathrm{Gap} = \frac{W_{\mathrm{FEA}}(y^{(s)}) - W_{\mathrm{FEA}}(y^{(f)})}{W_{\mathrm{FEA}}(y^{(f)})} \times 100\%.$$

In addition, the relative deviations of the six internal design parameters were also computed. Table 4 summarizes the validation statistics.

Table 4. Direct FEA validation statistics for the fidelity of the surrogate-optimized dataset.

Metric	Value
Number of validated scenarios	100
Mean optimality gap (%)	1.18
Median optimality gap (%)	0.92
95th percentile gap (%)	3.65
Maximum gap (%)	5.24
Proportion with gap $< 1\%$	54.0
Proportion with gap $< 2\%$	82.0
Mean relative deviation of the six internal parameters (%)	6.85

The results show that the surrogate-guided labels remain close to the direct-FEA reference optima. For the validated subset, the mean optimality gap is 1.18%, the median gap is 0.92%, and the 95th-percentile gap is 3.65%. In total, 54% of the samples exhibit a gap below 1%, and 82% exhibit a gap below 2%, while the worst-case gap is 5.24%. The mean relative deviation over the six internal design parameters is 6.85%. These results indicate that, although the Stage 2 dataset is generated on the emulator-defined fitness landscape, its labels are near-optimal with respect to direct finite-element re-optimization for the sampled scenarios.

The larger relative deviations observed in parameter space, compared with the much smaller objective gaps, suggest the presence of multiple near-optimal parameter combinations with similar structural weight in the local design landscape. Therefore, the Stage 2 dataset is interpreted in this study as a large-scale surrogate-optimized dataset with validated near-optimal fidelity. This validation supports its use as a high-quality label source for training the final Design Generator, while also preserving a clear distinction between surrogate-guided optima and exact direct-FEA optima.

From the perspective of error propagation, the present results indicate that the two-stage approximation does not invalidate the reliability of the final design generator within its intended scope. Although the Stage 2 emulator error can propagate into the Stage 2 optimization labels, the direct-FEA subset validation shows that the resulting objective-level deviation remains limited for representative sampled scenarios. Accordingly, the downstream generator should be interpreted as learning a validated near-optimal surrogate-guided design mapping rather than an exact direct-FEA optimum operator. Its reliability, therefore, lies in generating mechanically admissible and weight-efficient preliminary design candidates with controlled approximation error, rather than in guaranteeing exact finite-element global optima for every design instance.

4.4. Surrogate Model Performance Validation

After generating the surrogate-optimized design dataset and validating its near-optimal fidelity on a representative direct-FEA subset, the third stage involves training the final design generator. To evaluate the benefit of the physics-regularized formulation, a conventional MLP model was trained on the same dataset with the same objective: learning the mapping from 12 external conditions to the corresponding surrogate-optimized internal design parameters.

This section rigorously compares the performance of the physics-regularized design generator against the MLP-based design generator. The analysis is carried out from three critical perspectives: prediction accuracy on the full dataset, data efficiency under varying

sample sizes, and generalization capability in unseen design regions. Through quantitative metrics and visual comparisons, we demonstrate that the proposed generator not only achieves higher accuracy but also exhibits significantly better data adaptability and physical consistency, making it a more reliable tool for automated design.

4.4.1. Comparative Accuracy on the Full Dataset

Under the condition of the full dataset, the physics-regularized generator and the baseline MLP generator were trained separately, and their accuracy in reproducing the surrogate-optimized internal parameters was evaluated on an independent test set. The results are presented in Table 5.

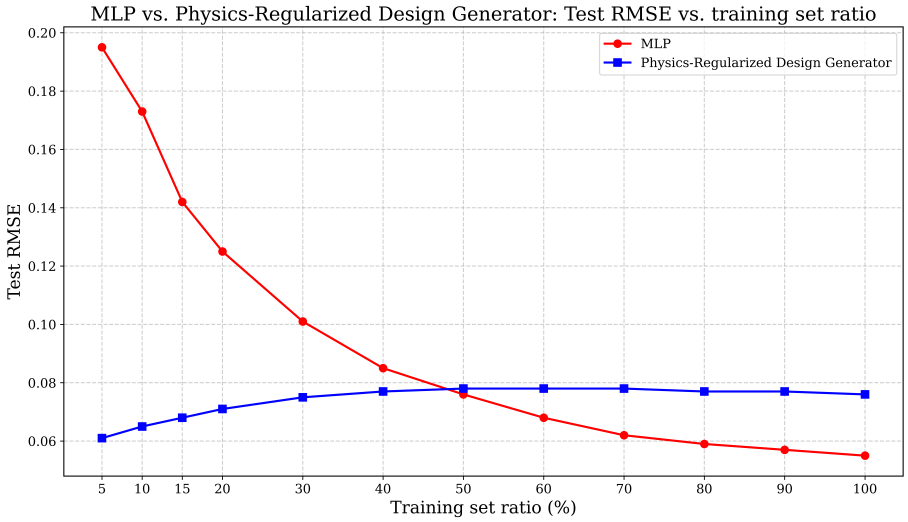
Table 5. Performance comparison of the physics-regularized generator and the MLP generator on the full dataset across all six output variables.

Output Variable	Model	MAE	RMSE	R^2
Rise-to-span ratio	MLP	0.0228	0.0359	0.964
	physics-regularized generator	0.0142	0.0218	0.987
Longitudinal grid size	MLP	0.1193	0.1472	0.948
	physics-regularized generator	0.0785	0.1036	0.981
Transverse grid size	MLP	0.1011	0.1283	0.952
	physics-regularized generator	0.0712	0.0924	0.986
Maximum shell thickness	MLP	0.0078	0.0116	0.976
	physics-regularized generator	0.0041	0.0063	0.993
Lower chord thickening ratio	MLP	0.0173	0.0257	0.961
	physics-regularized generator	0.0109	0.0146	0.989
Bending-moment redistribution index	MLP	0.0267	0.0341	0.944
	physics-regularized generator	0.0155	0.0208	0.983

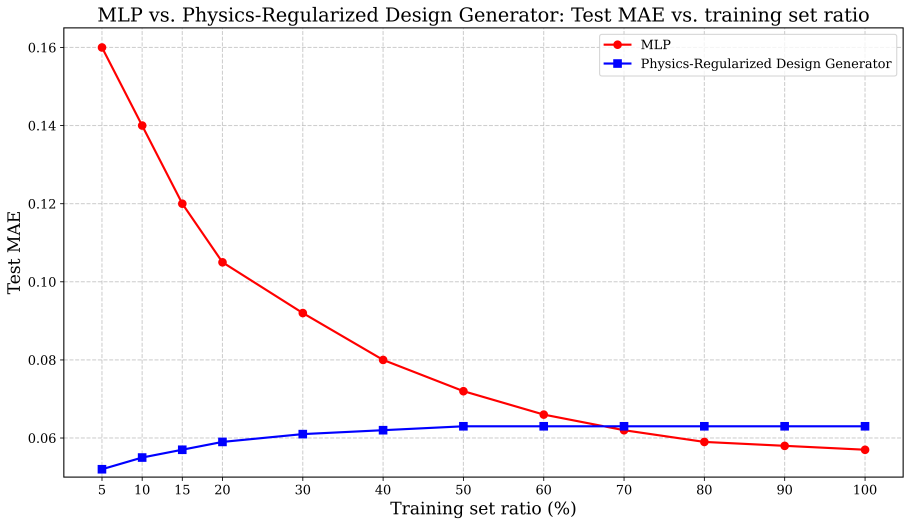
As can be seen from Table 5, when data is abundant, the physics-regularized design generator consistently outperforms the MLP generator across all six output parameters, achieving smaller errors (MAE, RMSE) and a higher goodness of fit (R^2). This indicates its superior ability to accurately learn and reproduce the complex mapping to the surrogate-optimized design solutions. Particularly for variables with strong physical significance and a major impact on structural economy, such as maximum shell thickness and rise-to-span ratio, the RMSE of the physics-regularized design generator is reduced by over 40% compared to the MLP. This result suggests that the introduction of physical energy principles as a regularizer enhances the generator's ability to capture mechanically meaningful trends in the design mapping, thereby improving its accuracy and reliability.

4.4.2. Comparative Data Efficiency

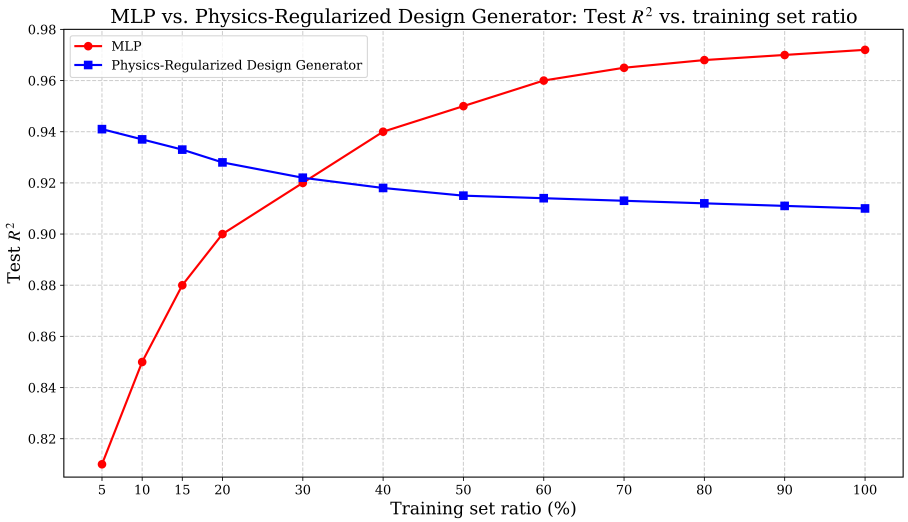
To further compare the adaptability and data utilization efficiency of the physics-regularized design generator and the MLP generator, especially under small-sample conditions, we trained both models under five different training sample proportions (100%, 50%, 20%, 10%, and 5%). Their performance was evaluated on a fixed test set, with the RMSE comparison shown in Table 6 and the full trend curves for all metrics plotted in Figure 7.



(**a**)



(**b**)



(**c**)

Figure 7. Comparative analysis of model accuracy variation with training data volume. (**a**) RMSE. (**b**) MAE. (**c**) R².

Table 6. Comparison of RMSE values for the MLP generator and the physics-regularized design generator under different training data fractions.

Training Data Fraction	MLP	Physics-Regularized Design Generator	Percentage Improvement
100%	0.0359	0.0218	+39.3
50%	0.0613	0.0271	+55.8
20%	0.0872	0.0338	+61.2
10%	0.1264	0.0427	+66.2
5%	0.1973	0.0609	+69.1

The results reveal a stark contrast in their sensitivity to data availability. The purely data-driven MLP generator demonstrates a high dependency on the volume of training data. As shown by its steep error curves (Figure 7a,b), both its RMSE and MAE metrics deteriorate sharply as the dataset shrinks, indicating its limited ability to learn the complex optimal mapping in data-scarce scenarios. This is further confirmed by its R^2 value (Figure 7c), which drops significantly, reflecting its failure to establish an effective fit without sufficient samples.

In contrast, the physics-regularized generator exhibits remarkable stability and data efficiency. Its error curves remain relatively flat across all data fractions, showcasing strong resistance to overfitting. This is because the embedded physical constraints act as a powerful regularizer, guiding the model toward mechanically plausible solutions that compensate for the lack of data coverage. Consequently, the physics-regularized generator maintains a consistently high R^2 value, and at the 5% data mark, where the performance gap is maximized, the physics-regularized generator's RMSE is merely 31% of the MLP's. This result indicates that the model learns a more stable and mechanically guided mapping than the purely data-driven baseline, rather than merely memorizing discrete data points, which makes it more suitable for practical engineering settings where generating massive surrogate-optimized datasets can be prohibitively expensive.

4.4.3. Comparative Generalization Capability

A trustworthy design generator must not only accurately reproduce known reference solutions but also generate physically plausible and reliable designs for entirely new, unseen combinations of external parameters. To rigorously validate this capability, we constructed two independent test sets corresponding to interpolation and extrapolation regimes. It is imperative to clarify the source of the "ground truth" for these validation samples, particularly in the extrapolation region. For each sample in the extrapolation set, a corresponding high-quality reference solution was regenerated under the same external conditions using an independent optimization procedure with direct FEA as the fitness evaluator throughout the search. These reference solutions were used as the benchmark for evaluating the extrapolation performance of the two generators. This strategy ensures that the assessment in the extrapolation region is grounded in direct-FEA-based benchmark solutions, rather than relying solely on surrogate-guided labels.

As illustrated in Figure 8, which visualizes performance in the interpolation region, both generators show good predictive trends. However, the scatter points for the physics-regularized generator are more densely clustered around the ideal diagonal line, indicating superior consistency and a more concentrated error distribution when generating solutions for new tasks that lie between known data points.

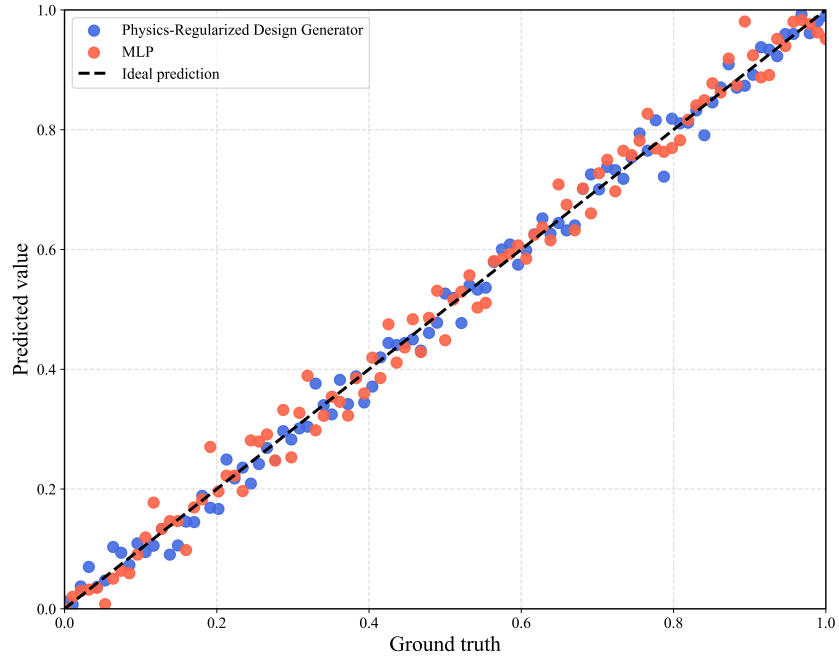


Figure 8. Visual comparison of generalization capability in the interpolation region.

Figure 9 provides a more critical assessment via heatmaps of prediction errors in the extrapolation region, revealing a stark contrast between the two models. The error distribution of the physics-regularized generator (top row of Figure 9) exhibits physically coherent, low-frequency patterns. This suggests that even when the generated parameters deviate from the true optimum, the deviation follows a systematic, physically plausible trend, ensuring the generated design remains robust and rational. Conversely, the MLP generator's error distributions (bottom row of Figure 9) are characterized by high-frequency, stochastic patterns, resembling white noise in several plots. This classic symptom of a "black-box" model failing to generalize means it is likely to generate erratic, unreliable, and physically inconsistent designs in unseen regions. This visual evidence provides a compelling case: the physics-regularized design generator not only achieves significantly lower errors but also maintains the physical consistency and interpretability of its outputs, a critical requirement for any automated design tool intended for real-world engineering.

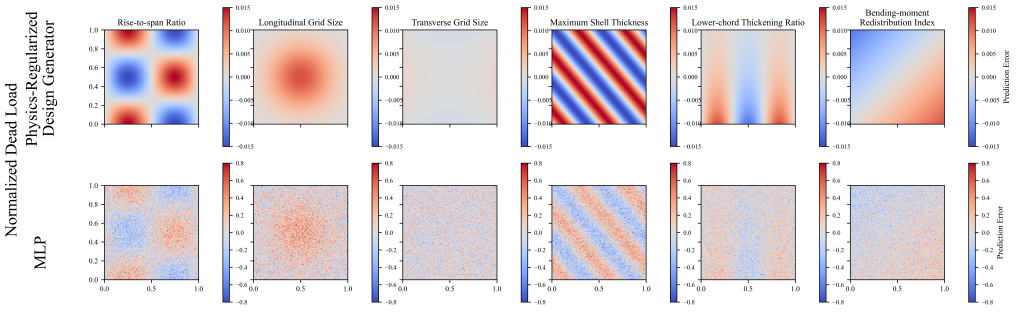


Figure 9. Heatmaps of model prediction errors in the extrapolation region.

4.5. Final Validation of the Integrated Framework via a Case Study

After validating the performance of the physics-regularized design generator in the previous section, this final section evaluates the practical effectiveness of the integrated framework through a representative case study. The purpose of this case study is twofold: first, to demonstrate the millisecond-level generation of internal design parameters for a new project instance; second, to assess the structural economy of the generated design

through a like-for-like comparison under identical external input parameters and wind-related constraints.

To initiate the case study, one set of 12 external input parameters was specified to represent a new engineering project. These external conditions were then kept fixed for both the Baseline Design and the Optimized Design. Under this identical set of project constraints, the trained physics-regularized design generator instantaneously produced the corresponding 6 internal design parameters for the Optimized Design. In this way, the subsequent comparison isolates the effect of the internal design variables and avoids interference from changes in load, boundary, or geometric input conditions.

4.5.1. Comparative Analysis of the Generated Design

To demonstrate the engineering value of the automated design framework, the Baseline Design and the Optimized Design are evaluated under the same 12 external input parameters, including geometric constraints, loads, wind-related factors, and boundary conditions. The difference between the two schemes is limited to the 6 internal design parameters.

The Baseline Design adopts a conventional parameter combination based on engineering practice, whereas the Optimized Design uses the 6 internal parameters generated by the trained physics-regularized model under the same external conditions. Table 7 summarizes the common external conditions, the two sets of internal design parameters, and the resulting total steel weight.

Table 7. Comparison between the Baseline Design and the Optimized Design.

Category	Parameter (Units)	Baseline	Optimized
External input parameters	Structural span (m)	120	120
	Structural length (m)	270	270
	Panels between supports	5	5
	Minimum shell thickness (m)	0.5	0.5
	Permissible void ratio of lower chord	0.5	0.5
	Dead load (kN/m^2)	0.5	0.5
	Snow load (kN/m^2)	0.5	0.5
	End-wall influence mesh count	5	5
	Longitudinal support stiffness (kN/m)	90,000	90,000
	Basic wind pressure (kN/m^2)	0.45	0.45
	Wind-vibration factor	1.5	1.5
	Maximum crown elevation (m)	20	20
Internal design parameters	Rise-to-span ratio	0.25	0.28
	Longitudinal grid size (m)	6.0	6.4
	Transverse grid size (m)	6.0	6.1
	Maximum shell thickness (m)	1.5	2.6
	Lower chord thickening ratio	0.5	0.75
	Bending-moment redistribution index	0.5	0.85
Performance metric	Total steel weight	1250 t	986.4 t

The comparison in Table 7 shows that, under identical external input parameters and wind-related constraints, the total steel weight decreases from 1250 t for the Baseline Design to 986.4 t for the Optimized Design, corresponding to a reduction of 21.1%. Since the two schemes differ only in the six internal design parameters, this reduction directly reflects

the contribution of the generated parameter combination to structural economy under the same project conditions. This comparison was intentionally designed to isolate the effect of the internal design variables. Therefore, the reported weight reduction should be interpreted as the performance gain attributable to the generated internal parameter combination, rather than as a consequence of altered external requirements, loading conditions, or boundary settings.

A closer examination of the internal parameters indicates that the reduction in steel consumption is achieved through coordinated adjustment rather than through a uniform decrease in material-related variables. Compared with the Baseline Design, the Optimized Design adopts a higher rise-to-span ratio, increasing from 0.25 to 0.28. This change implies a steeper structural form and is generally associated with more efficient force transfer along the shell surface, which is beneficial for improving the global load-carrying mechanism of the dome. At the same time, the longitudinal and transverse grid sizes increase from 6.0 m to 6.4 m and 6.1 m, respectively. This moderate enlargement of the grid indicates a tendency toward a more economical structural subdivision, which can reduce the total number of members and joints while maintaining the overall geometric continuity of the reticulated shell.

In parallel with these global geometric adjustments, the optimized solution also strengthens several stiffness-related internal parameters. The maximum shell thickness increases from 1.5 m to 2.6 m, the lower chord thickening ratio increases from 0.50 to 0.75, and the bending-moment redistribution index increases from 0.50 to 0.85. These changes indicate that the generated design does not seek weight reduction by uniformly weakening the structure. Instead, it combines a more efficient overall structural form with stronger local load-resisting capacity in critical regions. From an engineering perspective, this parameter pattern suggests a design strategy in which material is reduced where global efficiency can be improved, while stiffness and reserve capacity are enhanced where force redistribution is more important.

It is also noteworthy that several internal parameters increase substantially, whereas the total steel weight still decreases by 21.1%. This result implies that the optimized design improves material utilization at the system level rather than relying on simple dimensional reduction. In other words, the generated parameter combination reflects a balance between global form efficiency and local strengthening demand. For the case considered here, this balance leads to a structurally more economical solution than the conventional baseline configuration under the same external conditions.

Overall, the results demonstrate that the proposed framework can generate an internal design scheme with clear engineering rationality at the preliminary design stage. The optimized solution is characterized not by a single dominant variable, but by the joint adjustment of geometric and stiffness-related parameters, which together produce a measurable reduction in structural steel consumption under the same prescribed project conditions. At the same time, this case study should be interpreted as evidence of improved candidate-scheme quality rather than as a complete substitute for final structural acceptance. In practical application, the generated parameter combination should still be followed by project-specific high-fidelity analysis, member-level checking, and relevant code-based review before engineering implementation. Figure 10 further complements the parameter-level comparison by visualizing the stress distribution and thickness allocation of the Baseline Design and the Optimized Design under the same external conditions. As shown in Figure 10, the optimized scheme achieves weight reduction through coordinated material redistribution and improved structural response, rather than through indiscriminate weakening of the structural system.

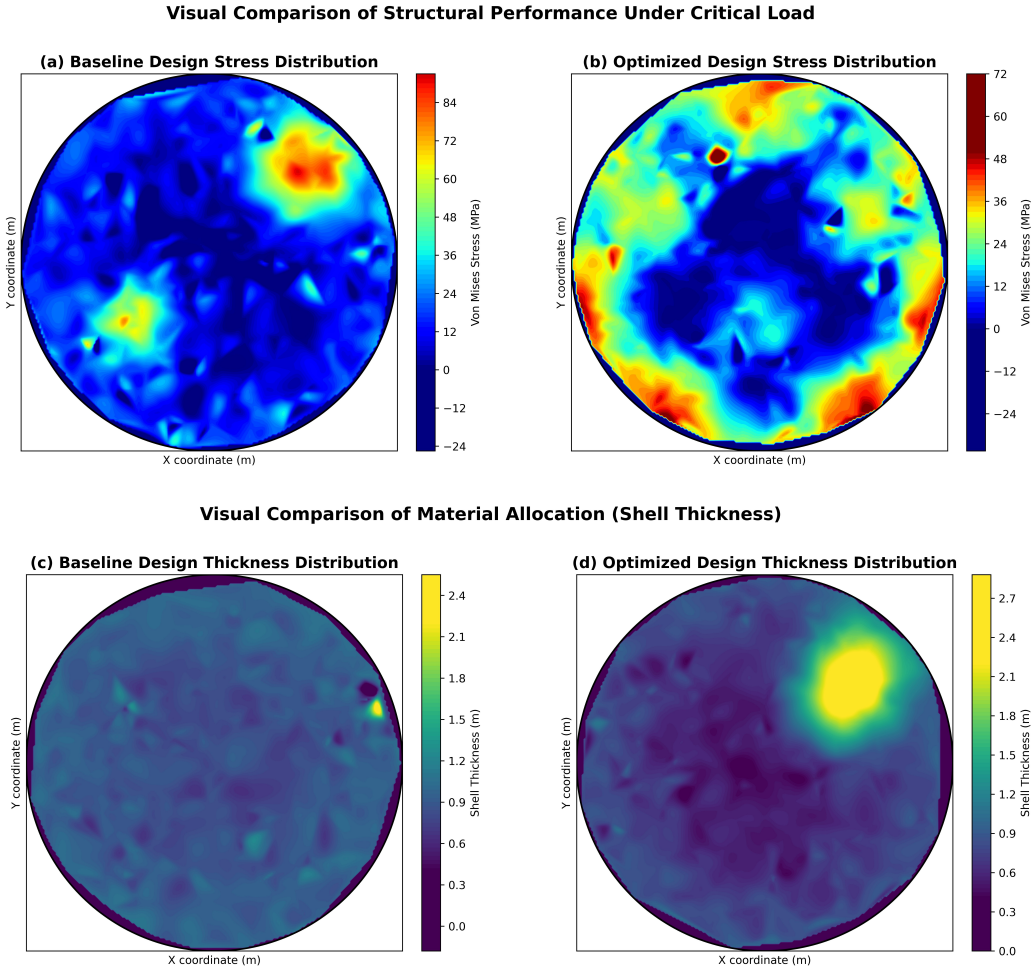


Figure 10. Visual comparison of the Baseline Design and the Optimized Design under identical external input parameters.

4.5.2. Analysis of Framework Efficiency and Value Proposition

The primary motivation for this work is to address the significant computational expense of traditional "FEA-in-the-loop" optimization. This section analyzes the efficiency of the proposed "Surrogate-in-the-loop" framework, focusing on how it restructures the computational workflow from a recurring cost model to one based on a reusable asset.

The framework's main computational expense is the one-time, upfront investment to generate the dataset and train the physics-regularized model, which involved a comprehensive data generation and training campaign spanning approximately three months of continuous high-fidelity computing. While this duration is non-trivial, it is crucial to recognize that this process creates a durable and reusable digital asset: a high-fidelity surrogate model that encapsulates the structural behavior across the entire design space.

Once trained, this asset can be deployed for subsequent optimization tasks at a negligible marginal cost. To quantify the strategic advantage of this framework, the upfront time investment is compared with the recurring timeline of conventional engineering design. In practice, a single FEA-in-the-loop optimization project for a complex long-span structure—considering model preparation, iterative simulation, and convergence checks—typically requires approximately 2 to 4 weeks. On this basis, the present three-month upfront investment reaches time-cost parity after approximately 3 to 5 full design projects. This estimate is intended at the project level rather than at the level of individual internal iterations or sub-tasks. Once this break-even point is passed, every subsequent project can benefit from near-instantaneous candidate generation, transforming the upfront

computational campaign into a reusable design asset rather than a recurring computational burden. Furthermore, a marginal benefit analysis reveals the strategic efficiency of the proposed approach. While the training of a conventional MLP requires continuous massive data updates, the physics-regularized 'Optimality Operator' internalizes the physical laws, allowing the framework to maintain high accuracy even if the sample size were reduced by up to 90% (as demonstrated in Section 4.4.2). This highlights that the three-month computational campaign for 100,000 samples represents a conservative, high-safety investment aimed at maximum fidelity; for scenarios requiring faster deployment, the physics-regularized nature of the final design generator allows for a substantial reduction in the initial data requirement without a proportional loss in reliability.

However, the framework's value extends beyond this long-term economic calculation. Its primary advantages are realized immediately:

1. Accelerated Design Iteration: It substantially shortens the project-level design-analysis cycle, enabling greater agility in responding to design changes and exploring alternatives.
2. Expanded Design Exploration: The low cost per run facilitates comprehensive exploration of the design space, increasing the potential to find non-intuitive, high-performance solutions that would otherwise be missed due to computational constraints.
3. Enhanced Reliability: The physics-regularized design generator guides the learning process using mechanically grounded principles, thereby increasing confidence in the generated preliminary design schemes.

In summary, the efficiency of the proposed framework is twofold. It offers a clear long-term economic advantage, and more importantly, it provides immediate strategic value by enhancing the speed, scope, and reliability of the entire design optimization process.

5. Discussion

This chapter aims to provide a deeper interpretation of the preceding experimental results, explore the core mechanisms behind the performance advantages of the proposed automated optimization framework, articulate its theoretical and practical significance for advancing automated design processes in the construction industry, and identify the limitations of the current research and potential directions for future work.

5.1. Interpretation of Key Findings

5.1.1. The Strategic Power of Decoupling

A central finding of this research is that the framework's success is not merely due to the application of a superior model, but to the strategic architecture of the entire multi-stage workflow. The cornerstone of this architecture is the decoupling of two fundamentally different tasks: rapid performance prediction and reliable optimal design generation. This separation allowed for a "division of labor," where different computational tools were deployed for the tasks they are best suited for, creating a synergistic effect that would be impossible to achieve with a monolithic approach.

First, the framework strategically offloads the computationally intensive, "brute-force" task of design space exploration to the most efficient tool for the job: a simple, high-speed MLP serving as an FEA Emulator. As validated in Section 4.2, this emulator provided near-instantaneous performance predictions with extremely high fidelity. This enabled the Genetic Algorithm to perform a massive optimization campaign—executing 100,000 independent searches—a task that would be computationally infeasible using direct FEA-in-the-loop optimization and inefficient with more complex approaches such as PINN-as-solver frameworks. The emulator's sole purpose was to act as a disposable, high-speed engine for creating the surrogate-optimized design dataset. Furthermore, the strategic decoupling of exploration and learning ensures that the framework's validity is not tethered

to a specific optimization technique. By framing the Stage 2 process as an interchangeable 'data-generation service,' the framework offers a future-proof architecture where more advanced or specialized optimizers can be integrated to further accelerate the asset-creation phase without altering the final physics-regularized generation logic.

Second, this two-stage data strategy improves the information quality of the training set by replacing randomly scattered design samples with a curated dataset of surrogate-optimized and well-converged labels. The final physics-regularized design generator therefore learns from a structured approximation of the high-performance design region rather than from heterogeneous designs of mixed quality. At the same time, it should be noted that the Stage 2 labels are obtained through GA optimization driven by the emulator, not by direct FEA-in-the-loop optimization. For this reason, the learned mapping should be interpreted as an approximation to the surrogate-optimized design landscape. The additional direct FEA subset validation shows that these labels possess near-optimal fidelity for representative sampled scenarios, but it does not imply that the entire 100,000-sample dataset consists of exact finite-element optima. Instead, the final generator should be understood as learning a validated near-optimal design mapping with controlled approximation error, which is suitable for rapid preliminary design generation but still requires project-specific high-fidelity verification before engineering adoption. The high predictive accuracy of the emulator and the small validated optimality gaps together support the engineering usefulness of this strategy, while residual surrogate-induced label error remains a meaningful topic for future refinement. By learning directly from a dataset of high-quality surrogate-optimized outcomes, the generator can more effectively capture the structurally meaningful trends associated with high-performance designs [70,71].

In summary, this strategic decoupling is the core mechanism that makes the entire framework both computationally tractable and highly effective. It intelligently manages computational resources by using a "low-cost" tool for a "high-volume" task (data generation) and a "high-value" tool for a "high-quality" task (learning optimal principles), proving to be a superior strategy to using a single, one-size-fits-all model.

5.1.2. The Physics-Regularized Generator as a Reliable Learner

The experimental results presented in Section 4.4 demonstrate that the physics-regularized design generator is superior to its purely data-driven MLP counterpart in accuracy, data efficiency, and generalization. The fundamental reason for this superiority lies not only in the model architecture, but also in the learning mechanism introduced by physics regularization. Instead of treating design generation as a purely statistical curve-fitting task, the proposed model reformulates it as a function approximation problem constrained by structural mechanics, thereby providing a more stable inductive bias in both sparse-data and unseen-design regions.

This addresses a fundamental question regarding the necessity of physics-informed learning when high-quality surrogate-optimized data are available. As substantiated by the comparative analysis in Section 4.4, data quality alone does not guarantee a robust learned model. The baseline MLP, despite being trained on 100% of the available surrogate-optimized labels, produced stochastic error patterns in the extrapolation heatmaps, which is a classic symptom of a black-box model failing to capture the underlying design logic. By contrast, the physics-regularized design generator benefits from an energy-based regularization mechanism that guides the learning process toward mechanically plausible trends, allowing it to maintain strong performance even when training data are limited.

In this sense, the superiority of the physics-regularized design generator stems from the composite loss function, in which the physics term acts as a structurally meaningful regularizer. Unlike conventional parameter regularization, the present energy-based term

imposes prior mechanical structure on the learned mapping and restricts the function space toward mechanically admissible regions. This physical constraint serves as an inductive bias, particularly when interpolating between data points or extrapolating beyond the training distribution. It helps explain the smoother and more structurally coherent error patterns observed in Section 4.4.3, as well as the stronger data efficiency reported in Section 4.4.2. Overall, the physics-regularized design generator can therefore be understood as a more reliable learner for this task because it combines empirical supervision with mechanical guidance, resulting in improved robustness, interpretability, and engineering trustworthiness relative to the purely data-driven baseline.

5.1.3. The Paradigm Shift from "Solving" to "Asset Creation"

Beyond the technical performance of the individual components, the proposed framework embodies a fundamental paradigm shift in the application of computational tools for automated design: a shift from a recurring "per-task solving" model to a "one-time asset creation" model. This change has profound implications for the efficiency, economics, and scalability of engineering design workflows.

The traditional "FEA-in-the-loop" paradigm, and even the more recent "PINN-as-solver" approach, treat optimization as a service. For each new design problem, a computationally expensive solving process is initiated from scratch, incurring a high, recurring cost in terms of time and resources. The computational effort is directly proportional to the number of design tasks undertaken.

Our multi-stage framework fundamentally restructures this economic model. It consolidates the prohibitive computational cost into a significant, but one-time, upfront investment to generate the necessary datasets and train the final physics-regularized design generator. The outcome of this investment is not merely a single optimized solution, but a durable and reusable digital asset. As the efficiency analysis in Section 4.5.2 quantitatively indicates, this front-loaded investment is recouped after approximately 3 to 5 full design projects, depending on the benchmark duration adopted for a conventional optimization project. Beyond that point, the framework yields progressively greater returns in terms of time and computational efficiency.

Once created, this digital asset encapsulates the complex physical knowledge of the entire design space and can be deployed to solve an unlimited number of new design problems within its domain at a negligible marginal cost. Its prediction speed is instantaneous and, crucially, independent of the complexity of the original FEA model. This decouples the complexity of the physical simulation from the iterative design exploration process. This transformation from a recurring, service-like cost model to a reusable asset model is a central contribution of this work. It suggests a more strategic approach where engineering firms can invest in creating proprietary, high-fidelity design assets, fundamentally altering the landscape of how design exploration and optimization are performed in the AEC industry.

5.2. Implications for Automated Design Processes

The proposed design generation framework is not merely a technical enhancement; it represents a potential reconstruction of the traditional design workflow in the AEC industry. Its implications are best understood across three key dimensions:

1. Workflow Transformation: The framework fundamentally inverts the traditional "human-waits-for-machine" paradigm. By providing an instantaneous "machine-serves-human" feedback loop, it transforms the design process from a slow, serial task into a real-time, interactive exploration. This liberates engineers from tedious computational waiting, allowing them to focus on higher-level creative and strategic tasks.

2. Enhanced Design Exploration: By enabling the instant generation of countless high-quality solutions, the tool evolves from a mere "optimizer" of existing ideas into a true "generator" of novel ones. This vastly expands the scope of design exploration, increasing the probability of discovering non-intuitive, high-performance solutions that lie beyond the bounds of conventional experience.
3. Trustworthy Automation: Crucially, unlike purely black-box generators that may produce physically implausible pseudo-optimal solutions, the physics-regularized design generator introduces a mechanics-based inductive bias that improves the reliability of its outputs at the preliminary design stage. This reliability should be understood in the sense of generating mechanically more admissible and structurally more rational candidate schemes within the learned design domain, rather than in the sense of replacing explicit checks on stress limits, displacement limits, stability, constructability, or code compliance. Accordingly, the framework should be viewed as a trustworthy front-end design generation tool that reduces the risk of mechanically inconsistent candidates and improves the efficiency of subsequent verification workflows.

This framework paves the way for a more agile and intelligent design process, shifting the designer's role from computational executor to strategic creator.

5.3. Limitations and Future Work

Although the proposed automated optimization framework has demonstrated significant advantages, it is necessary to acknowledge several limitations of the current study, which in turn point to clear directions for future research.

First, a critical boundary condition of the current framework is its reliance on a linear-elastic reduced-order energy regularization. The physics term embedded in the design generator is formulated from a parameterized equivalent stiffness model and is therefore primarily targeted at the Serviceability Limit State (SLS), where global stiffness and deflection-related behavior are dominant concerns. Although this choice preserves millisecond-level inference and avoids online finite-element analysis during training, it does not explicitly enforce member-level stress limits, code-defined displacement checks, local or global buckling resistance, second-order effects, connection detailing requirements, or broader constructability constraints. These aspects remain indispensable in practical structural design and should not be inferred as being fully resolved by the present objective function. Accordingly, any generated design should still be treated as a mechanically informed preliminary candidate and must be subjected to project-specific high-fidelity finite-element verification, member checking, and relevant code-compliance review before engineering adoption. Future research will focus on extending the present reduced-order potential-energy formulation toward second-order and nonlinear stiffness contributions, and on incorporating explicit multi-constraint design criteria so as to support more direct ULS-oriented and code-aware generative design.

Second, the validation case in this study is limited to a specific structural type—the three-centered reticulated dome. Although the multi-stage framework is theoretically universal, its applicability and performance on other complex structural systems, such as free-form shells or cable-net structures, have not yet been verified.

Additionally, the current data generation strategy relies on a static, high-volume sampling approach. We acknowledge that the computational efficiency of the framework could be further enhanced by integrating active learning (AL), adaptive sampling, or multi-fidelity (MF) strategies. Such techniques could dynamically identify high-gradient regions of the design space and direct FEA resources more efficiently, potentially reducing the required upfront simulations by orders of magnitude. Although these advanced sampling methodologies were outside the immediate scope of this feasibility study—which focuses

on the verification of the physics-regularized design generator as a design generator—they represent critical avenues for future work to lower the entry barrier for applying this technology to even more complex structural systems.

These limitations naturally lead to several promising avenues for future work:

1. Integration of Non-Linear Physics: A crucial avenue for future work is to extend the framework by integrating non-linear governing equations into the physics-regularized design generator's loss function. This would enable the surrogate model to accurately predict and optimize for structural behavior at the ultimate limit state, thus supporting a more comprehensive, performance-based design methodology.
2. Application to Diverse Structural Systems: Future research should focus on extending and applying this automated framework to a more diverse range of structural types. This will require establishing corresponding parametric models and selecting or deriving appropriate governing physical equations for different structures, thereby testing and enhancing the framework's generality and flexibility.
3. Extension to Multi-Objective Optimization: Real-world engineering design is often a complex trade-off between multiple, conflicting objectives, such as economy, safety, and sustainability. Therefore, an important future direction is to upgrade the current single-objective framework (minimizing steel consumption) to a multi-objective one. This could be achieved by replacing the GA with multi-objective evolutionary algorithms (e.g., NSGA-II) and extending the physics-regularized design generator to predict a set of Pareto-optimal solutions, thereby providing designers with a range of high-performance trade-off options for more intelligent design decision support.

6. Conclusions

To address the dual challenges in the optimal design of complex long-span structures—namely, the impracticality of element-level optimization and the prohibitively high computational cost of traditional FEA workflows coupled with the fundamental deficiencies of standard surrogate models—this study has successfully proposed and validated a novel automated optimization framework. The framework's core begins with a fundamental problem reformulation: elevating the optimization to a system-level via a parameter system. Building on this, it architects a paradigm shift from a recurring "per-task solving" model to a "one-time asset creation" model, culminating in a reusable physics-regularized design generator.

The core mechanism of this framework lies in a multi-stage workflow that intelligently decouples performance prediction from optimal design generation. The proposed framework combines a high-speed MLP-based FEA Emulator, a GA-driven surrogate optimization stage, and a physics-regularized design generator for rapid structural design generation. The Stage 2 dataset is generated on the emulator-defined fitness landscape and further validated on a representative subset through direct finite-element re-optimization, showing that the resulting labels possess near-optimal fidelity for the sampled scenarios. This validated surrogate-optimized dataset then enables the final generator to learn an efficient mapping from external design conditions to high-quality internal parameter combinations.

The core contributions of this research can be summarized as follows:

1. A parameterized system-level optimization approach: This research introduces a system-level parametric modeling approach that elevates the optimization problem from the impractical 'element-level' to the efficient 'system-level'. This strategy of reformulating the problem at its source provides a conceptually feasible and computationally efficient foundation for the automated design of complex structures, serving as the cornerstone upon which the entire framework is successfully built.

2. A strategic multi-stage framework: We proposed and validated a robust, four-stage workflow that strategically assigns different roles to different AI models. This decoupled approach was proven to be a computationally tractable and highly effective method for creating a large-scale surrogate-optimized dataset with validated near-optimal fidelity, a critical step that is often a bottleneck in data-driven engineering.
3. A physics-regularized design generator: Through systematic comparative experiments, we demonstrated that the physics-regularized design generator consistently outperforms its purely data-driven MLP counterpart. It not only exhibits higher accuracy on the full dataset, but also maintains excellent performance under limited training data, achieving an error only 31% of that of the MLP model when trained on 5% of the data, and shows substantially superior generalization in previously unseen design regions, thereby improving reliability relative to purely black-box models.
4. Significant and verifiable performance gains: In a case study of a three-centered reticulated dome, the framework completed the generation of internal design parameters in milliseconds, whereas the conventional workflow would require iterative optimization. Under identical external input parameters and wind-related constraints, the generated design achieved a 21.1% reduction in total steel weight relative to the baseline configuration.

This study confirms that strategically integrating fundamental principles of physics into advanced machine learning workflows is a promising pathway for advancing design automation and intelligence in the construction industry. The proposed framework provides an effective front-end tool for rapidly generating weight-efficient and mechanically informed candidate schemes under prescribed project conditions. At the same time, its current role should be understood within the preliminary design stage rather than as a replacement for final structural verification, code-based checking, and engineering detailing. In this sense, the broader value of the framework lies in improving the quality and efficiency of early design exploration while reducing the likelihood of mechanically inconsistent candidate solutions. By shifting the focus from simply accelerating individual simulations to creating durable, reusable digital assets, this work offers a practical foundation for more efficient and trustworthy automated design workflows in future engineering applications.

Author Contributions: Conceptualization, X.C., G.Q. and J.G.; methodology, X.C.; software, X.C., S.S. and Y.Z.; validation, X.C.; formal analysis, X.C.; investigation, X.C.; resources, X.C.; data curation, X.C., S.S. and Y.Z.; writing—original draft preparation, X.C.; writing—review and editing, X.C., G.Q., J.G., S.S. and Y.Z.; visualization, X.C.; supervision, G.Q. and J.G.; project administration, X.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data presented in this study are available on request from the corresponding author.

Acknowledgments: The author sincerely thanks the supervising professors and senior colleagues in the research team for their invaluable guidance and selfless assistance throughout this research.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Cheng, K.-T.; Olhoff, N. An investigation concerning optimal design of solid elastic plates. *Int. J. Solids Struct.* **1981**, *17*, 305–323. [CrossRef]
2. Bendsøe, M.P.; Kikuchi, N. Generating optimal topologies in structural design using a homogenization method. *Comput. Methods Appl. Mech. Eng.* **1988**, *71*, 197–224. [CrossRef]
3. Svanberg, K. The method of moving asymptotes—A new method for structural optimization. *Int. J. Numer. Methods Eng.* **1987**, *24*, 359–373. [CrossRef]

4. Mai, H.T.; Kang, J.; Lee, J. A machine learning-based surrogate model for optimization of truss structures with geometrically nonlinear behavior. *Finite Elem. Anal. Des.* **2021**, *196*, 103572. [CrossRef]
5. Liu, J.; Xia, Y. A hybrid intelligent genetic algorithm for truss optimization based on deep neutral network. *Swarm Evol. Comput.* **2022**, *73*, 101120. [CrossRef]
6. White, D.A.; Arrighi, W.J.; Kudo, J.; Watts, S.E. Multiscale topology optimization using neural network surrogate models. *Comput. Methods Appl. Mech. Eng.* **2019**, *346*, 1118–1135. [CrossRef]
7. Kodiyalam, S.; Gurumoorthy, R. Neural networks with modified backpropagation learning applied to structural optimization. *AIAA J.* **1996**, *34*, 408–412. [CrossRef]
8. Cybenko, G. Approximation by superpositions of a sigmoidal function. *Math. Control. Signals Syst.* **1989**, *2*, 303–314. [CrossRef]
9. Hornik, K. Approximation capabilities of multilayer feedforward networks. *Neural Netw.* **1991**, *4*, 251–257. [CrossRef]
10. Ma, F.; Zeng, Z. High-risk prediction localization: Evaluating the reliability of black box models for topology optimization. *Struct. Multidiscip. Optim.* **2020**, *62*, 3053–3069. [CrossRef]
11. Woldseth, R.V.; Aage, N.; Bærentzen, J.A.; Sigmund, O. On the use of artificial neural networks in topology optimisation. *Struct. Multidiscip. Optim.* **2022**, *65*, 294. [CrossRef]
12. Ates, G.C.; Gorguluarslan, R.M. Two-stage convolutional encoder-decoder network to improve the performance and reliability of deep learning models for topology optimization. *Struct. Multidiscip. Optim.* **2021**, *63*, 1927–1950. [CrossRef]
13. Islam, M.M.; Liu, L. Deep learning accelerated topology optimization with inherent control of image quality. *Struct. Multidiscip. Optim.* **2022**, *65*, 325. [CrossRef]
14. Karniadakis, G.E.; Kevrekidis, I.G.; Lu, L.; Perdikaris, P.; Wang, S.; Yang, L. Physics-informed machine learning. *Nat. Rev. Phys.* **2021**, *3*, 422–440. [CrossRef]
15. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [CrossRef]
16. Cuomo, S.; Di Cola, V.S.; Giampaolo, F.; Rozza, G.; Raissi, M.; Piccialli, F. Scientific machine learning through physics-informed neural networks: Where we are and what's next. *J. Sci. Comput.* **2022**, *92*, 88. [CrossRef]
17. Jeong, H.; Bai, J.; Batuwatta-Gamage, C.P.; Rathnayaka, C.; Zhou, Y.; Gu, Y. A physics-informed neural network-based topology optimization (PINNTO) framework for structural optimization. *Eng. Struct.* **2023**, *278*, 115484. [CrossRef]
18. Zehnder, J.; Li, Y.; Coros, S.; Thomaszewski, B. Ntopo: Mesh-free topology optimization using implicit neural representations. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 10368–10381.
19. Grossmann, T.G.; Komorowska, U.J.; Latz, J.; Schönlieb, C.-B. Can physics-informed neural networks beat the finite element method? *IMA J. Appl. Math.* **2024**, *89*, 143–174. [CrossRef]
20. Chi, H.; Zhang, Y.; Tang, T.L.E.; Mirabella, L.; Dalloro, L.; Song, L.; Paulino, G.H. Universal machine learning for topology optimization. *Comput. Methods Appl. Mech. Eng.* **2021**, *375*, 112739. [CrossRef]
21. Senhora, F.V.; Chi, H.; Zhang, Y.; Mirabella, L.; Tang, T.L.E.; Paulino, G.H. Machine learning for topology optimization: Physics-based learning through an independent training strategy. *Comput. Methods Appl. Mech. Eng.* **2022**, *398*, 115116. [CrossRef]
22. Lagaros, N.D.; Charmpis, D.C.; Papadrakakis, M. An adaptive neural network strategy for improving the computational performance of evolutionary structural optimization. *Comput. Methods Appl. Mech. Eng.* **2005**, *194*, 3374–3393. [CrossRef]
23. Papadrakakis, M.; Lagaros, N.D.; Tsompanakis, Y. Structural optimization using evolution strategies and neural networks. *Comput. Methods Appl. Mech. Eng.* **1998**, *156*, 309–333. [CrossRef]
24. Xie, Y.M.; Steven, G.P. A simple evolutionary procedure for structural optimization. *Comput. Struct.* **1993**, *49*, 885–896. [CrossRef]
25. Sigmund, O. On the usefulness of non-gradient approaches in topology optimization. *Struct. Multidiscip. Optim.* **2011**, *43*, 589–596. [CrossRef]
26. Yildiz, A.; Öztürk, N.; Kaya, N.; Öztürk, F. Integrated optimal topology design and shape optimization using neural networks. *Struct. Multidiscip. Optim.* **2003**, *25*, 251–260. [CrossRef]
27. Kirchdoerfer, T.; Ortiz, M. Data-driven computational mechanics. *Comput. Methods Appl. Mech. Eng.* **2016**, *304*, 81–101. [CrossRef]
28. Ulu, E.; Zhang, R.; Kara, L.B. A data-driven investigation and estimation of optimal topologies under variable loading configurations. *Comput. Methods Biomech. Biomed. Eng. Imaging Vis.* **2016**, *4*, 61–72. [CrossRef]
29. Sasaki, H.; Igarashi, H. Topology optimization accelerated by deep learning. *IEEE Trans. Magn.* **2019**, *55*, 7401305. [CrossRef]
30. LeCun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
31. LeCun, Y.; Bottou, L.; Bengio, Y.; Haffner, P. Gradient-based learning applied to document recognition. *Proc. IEEE* **2002**, *86*, 2278–2324. [CrossRef]
32. Abueidda, D.W.; Koric, S.; Sobh, N.A. Topology optimization of 2D structures with nonlinearities using deep learning. *Comput. Struct.* **2020**, *237*, 106283. [CrossRef]
33. Asanuma, J.; Doi, S.; Igarashi, H. Transfer learning through deep learning: Application to topology optimization of electric motor. *IEEE Trans. Magn.* **2020**, *56*, 7512404. [CrossRef]

34. Wang, D.; Xiang, C.; Pan, Y.; Chen, A.; Zhou, X.; Zhang, Y. A deep convolutional neural network for topology optimization with perceptible generalization ability. *Eng. Optim.* **2022**, *54*, 973–988. [CrossRef]
35. Yan, J.; Zhang, Q.; Xu, Q.; Fan, Z.; Li, H.; Sun, W.; Wang, G. Deep learning driven real time topology optimisation based on initial stress learning. *Adv. Eng. Inform.* **2022**, *51*, 101472. [CrossRef]
36. Deng, C.; Wang, Y.; Qin, C.; Fu, Y.; Lu, W. Self-directed online machine learning for topology optimization. *Nat. Commun.* **2022**, *13*, 388. [CrossRef]
37. Kallioras, N.A.; Kazakis, G.; Lagaros, N.D. Accelerated topology optimization by means of deep learning. *Struct. Multidiscip. Optim.* **2020**, *62*, 1185–1212. [CrossRef]
38. Banga, S.; Gehani, H.; Bhilare, S.; Patel, S.; Kara, L. 3D topology optimization using convolutional neural networks. *arXiv* **2018**, arXiv:1808.07440. [CrossRef]
39. Lee, S.; Kim, H.; Lieu, Q.X.; Lee, J. CNN-based image recognition for topology optimization. *Knowl.-Based Syst.* **2020**, *198*, 105887. [CrossRef]
40. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. ImageNet classification with deep convolutional neural networks. *Commun. ACM* **2017**, *60*, 84–90. [CrossRef]
41. Ronneberger, O.; Fischer, P.; Brox, T. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*; Springer: Berlin/Heidelberg, Germany, 2015; pp. 234–241.
42. Sosnovik, I.; Oseledets, I. Neural networks for topology optimization. *Russ. J. Numer. Anal. Math. Model.* **2019**, *34*, 215–223. [CrossRef]
43. Yu, B. The deep Ritz method: A deep learning-based numerical algorithm for solving variational problems. *Commun. Math. Stat.* **2018**, *6*, 1–12. [CrossRef]
44. Guo, T.; Lohan, D.J.; Cang, R.; Ren, M.Y.; Allison, J.T. An indirect design representation for topology optimization using variational autoencoder and style transfer. In Proceedings of the 2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference, Kissimmee, FL, USA, 8–12 January 2018; p. 0804.
45. Herath, S.; Haputhanthri, U. Topologically optimal design and failure prediction using conditional generative adversarial networks. *Int. J. Numer. Methods Eng.* **2021**, *122*, 6867–6887. [CrossRef]
46. Lin, Q.; Hong, J.; Liu, Z.; Li, B.; Wang, J. Investigation into the topology optimization for conductive heat transfer based on deep learning approach. *Int. Commun. Heat Mass Transf.* **2018**, *97*, 103–109. [CrossRef]
47. Oh, S.; Jung, Y.; Kim, S.; Lee, I.; Kang, N. Deep generative design: Integration of topology optimization and generative models. *J. Mech. Des.* **2019**, *141*, 111405. [CrossRef]
48. Rawat, S.; Shen, M. A novel topology design approach using an integrated deep learning network architecture. *arXiv* **2018**, arXiv:1808.02334.
49. Saha, S.; Gan, Z.; Cheng, L.; Gao, J.; Kafka, O.L.; Xie, X.; Li, H.; Tajdari, M.; Kim, H.A.; Liu, W.K. Hierarchical deep learning neural network (HiDeNN): An artificial intelligence (AI) framework for computational science and engineering. *Comput. Methods Appl. Mech. Eng.* **2021**, *373*, 113452. [CrossRef]
50. Haghighat, E.; Raissi, M.; Moure, A.; Gomez, H.; Juanes, R. A physics-informed deep learning framework for inversion and surrogate modeling in solid mechanics. *Comput. Methods Appl. Mech. Eng.* **2021**, *379*, 113741. [CrossRef]
51. Raissi, M.; Karniadakis, G.E. Hidden physics models: Machine learning of nonlinear partial differential equations. *J. Comput. Phys.* **2018**, *357*, 125–141. [CrossRef]
52. Baydin, A.G.; Pearlmutter, B.A.; Radul, A.A.; Siskind, J.M. Automatic differentiation in machine learning: A survey. *J. Mach. Learn. Res.* **2018**, *18*, 5595–5637.
53. Chandrasekhar, A.; Sridhara, S.; Suresh, K. Auto: A framework for automatic differentiation in topology optimization. *Struct. Multidiscip. Optim.* **2021**, *64*, 4355–4365. [CrossRef]
54. Raissi, M.; Yazdani, A.; Karniadakis, G.E. Hidden fluid mechanics: Learning velocity and pressure fields from flow visualizations. *Science* **2020**, *367*, 1026–1030. [CrossRef] [PubMed]
55. Hu, H.; Qi, L.; Chao, X. Physics-informed Neural Networks (PINN) for computational solid mechanics: Numerical frameworks and applications. *Thin-Walled Struct.* **2024**, *205*, 112495. [CrossRef]
56. Yang, J.; Huang, W.; Huang, Q.; Hu, H. An investigation on the coupling of data-driven computing and model-driven computing. *Comput. Methods Appl. Mech. Eng.* **2022**, *393*, 114798. [CrossRef]
57. Lu, L.; Jin, P.; Pang, G.; Zhang, Z.; Karniadakis, G.E. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nat. Mach. Intell.* **2021**, *3*, 218–229. [CrossRef]
58. Luo, J.; Li, Y.; Zhou, W.; Gong, Z.; Zhang, Z.; Yao, W. An Improved Data-Driven Topology Optimization. *CMES-Comput. Model. Eng. Sci.* **2021**, *128*, 823–848.
59. Jeong, H.; Batuwatta-Gamage, C.; Bai, J.; Rathnayaka, C.; Zhou, Y.; Gu, Y. An advanced physics-informed neural network-based framework for nonlinear and complex topology optimization. *Eng. Struct.* **2025**, *322*, 119194. [CrossRef]

60. Mai, H.T.; Mai, D.D.; Kang, J.; Lee, J.; Lee, J. Physics-informed neural energy-force network: A unified solver-free numerical simulation for structural optimization. *Eng. Comput.* **2024**, *40*, 147–170. [CrossRef]
61. Joglekar, A.; Chen, H.; Kara, L.B. DMF-TONN: Direct mesh-free topology optimization using neural networks. *Eng. Comput.* **2024**, *40*, 2227–2240. [CrossRef]
62. Zhang, Z.; Yao, W.; Li, Y.; Zhou, W.; Chen, X. Topology optimization via implicit neural representations. *Comput. Methods Appl. Mech. Eng.* **2023**, *411*, 116052. [CrossRef]
63. Zhang, Z.; Li, Y.; Zhou, W.; Chen, X.; Yao, W.; Zhao, Y. TONR: An exploration for a novel way combining neural network with topology optimization. *Comput. Methods Appl. Mech. Eng.* **2021**, *386*, 114083. [CrossRef]
64. He, J.; Chadha, C.; Kushwaha, S.; Koric, S.; Abueidda, D.; Jasiuk, I. Deep energy method in topology optimization applications. *Acta Mech.* **2023**, *234*, 1365–1379. [CrossRef]
65. Adeli, H.; Park, H.S. A neural dynamics model for structural optimization—Theory. *Comput. Struct.* **1995**, *57*, 383–390. [CrossRef]
66. Berke, L.; Hajela, P. Applications of artificial neural nets in structural mechanics. *Struct. Optim.* **1992**, *4*, 90–98. [CrossRef]
67. Hajela, P.; Berke, L. Neurobiological computational models in structural analysis and design. *Comput. Struct.* **1991**, *41*, 657–667. [CrossRef]
68. Lu, L.; Meng, X.; Mao, Z.; Karniadakis, G.E. DeepXDE: A deep learning library for solving differential equations. *SIAM Rev.* **2021**, *63*, 208–228. [CrossRef]
69. Samaniego, E.; Anitescu, C.; Goswami, S.; Nguyen-Thanh, V.M.; Guo, H.; Hamdia, K.; Zhuang, X.; Rabczuk, T. An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. *Comput. Methods Appl. Mech. Eng.* **2020**, *362*, 112790. [CrossRef]
70. McClenny, L.D.; Braga-Neto, U.M. Self-adaptive physics-informed neural networks. *J. Comput. Phys.* **2023**, *474*, 111722. [CrossRef]
71. Du, Z.; Ding, X.; Chen, H.; Liu, C.; Zhang, W.; Luo, J.; Guo, X. Optimal design of topological waveguides by machine learning. *Front. Mater.* **2022**, *9*, 1075073. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.