 DOI: 10.9744/duts.13.1.36-51

AN INTELLIGENT HYBRID FRAMEWORK FOR REDUCING COMPUTATIONAL COMPLEXITY IN HIGH-DIMENSIONAL OPTIMIZATION

Steven Gaillard[1] and I-Tung Yang[2]
[1] Program Studi Magister Teknik Sipil, Universitas Kristen Petra
[2] National Taiwan University of Science and Technology, Department of Civil and Construction Engineering, Taipei, Taiwan, 10607

[1] gacchuguts@gmail.com, [2] ityang@mail.ntust.edu.tw

ABSTRACT: *This study proposes a hybrid framework that integrates structural analysis, dimensionality reduction (DR), and metaheuristic algorithms to enhance the design optimization of steel structures. Structural responses are extracted using ETABS, and high-dimensional design data are reduced via PCA, t-SNE, and deep Autoencoders. Classifiers such as XGBoost, LightGBM, and CatBoost are employed as diagnostic tools to verify that the reduced latent space preserves feasibility boundaries. Optimization is then performed using Particle Swarm Optimization (PSO) and Symbiotic Organisms Search (SOS) to minimize structural weight while meeting code requirements. The framework is validated on benchmark functions and a 20-story braced steel frame, showing that SOS consistently outperforms PSO and that MSE-based Autoencoders achieve superior convergence and solution quality compared to linear or shallow methods. The results highlight DR-integrated metaheuristics as a scalable and effective approach for large-scale optimization. Future work will extend the framework to predictive models and more complex structural systems.*

Keywords: *structural design optimization, high-dimensional structural analysis, dimensionality reduction, classifier, metaheuristics*

1. INTRODUCTION

The optimization of steel structures is essential for delivering building systems that are safe, efficient, and cost-effective. Conventional structural design optimization (SDO) generally aims to minimize material consumption while ensuring compliance with strength and serviceability codes. Yet, modern engineering projects—particularly tall buildings and large-scale frameworks—pose high-dimensional optimization challenges, where the number of design variables and constraints increases rapidly. This dimensional growth results in substantial computational demands, making traditional metaheuristic approaches less effective.

Metaheuristic algorithms such as Particle Swarm Optimization (PSO) (Kennedy & Eberhart, 1995) and Symbiotic Organisms Search (SOS) (Cheng & Prayogo, 2014) are popular for their ability to explore nonlinear, multi-modal search spaces. However, when applied to problems with hundreds or thousands of variables, they often suffer from slow convergence, long runtimes, and difficulty avoiding local optima. The added requirement of repeatedly performing structural analysis in software like ETABS further amplifies computational costs within the optimization loop.

Dimensionality reduction (DR) methods have been introduced as a solution, enabling high-dimensional design spaces to be projected into lower-dimensional, more tractable representations. Techniques such as Principal Component Analysis (PCA) (Pearson, 1901), t-Distributed Stochastic Neighbor Embedding (t-SNE) (Van der Maaten & Hinton, 2008), and Uniform Manifold Approximation and Projection (UMAP) (McInnes et al., 2018) simplify the search process while attempting to retain key structural characteristics. More advanced approaches, such as Autoencoders, leverage deep learning to capture nonlinear relationships between original and reduced spaces more effectively (Hinton & Salakhutdinov, 2006). Despite these advantages, most prior studies have employed DR independently, with limited validation of how well the reduced space reflects the original optimization landscape.

To address these limitations, this study proposes a hybrid framework that integrates DR with metaheuristic optimization to handle high-dimensional structural design tasks. The framework incorporates machine learning diagnostics and re-optimization in latent space, aiming to reduce computational costs without sacrificing accuracy. Its effectiveness is tested on a 20-story steel frame, demonstrating potential for practical application in real-world structural optimization problems.

2. THEORY AND HYPOTHESES

2.1. Metaheuristics Algorithms in Structural Optimization

Metaheuristic algorithms have gained wide application in SDO because of their ability to tackle complex, nonlinear, and discrete problems where conventional gradient-based methods are often inadequate. PSO, in particular, is recognized for its simplicity and rapid convergence. It has been applied in various structural engineering tasks such as truss design, frame member sizing, and bracing configuration, consistently achieving weight reductions while satisfying code requirements (Kennedy & Eberhart, 1995; Kaveh & Mahdavi, 2014).

Nonetheless, the performance of PSO declines in high-dimensional search spaces, where the increase in design variables may lead to premature convergence or entrapment in local optima, particularly under nonlinear constraints (Gandomi et al., 2013). To address these issues, SOS was introduced as an alternative. Inspired by biological interactions such as mutualism, commensalism, and parasitism, SOS has demonstrated robust results in structural optimization involving large discrete design domains. Unlike PSO, SOS is parameter-free, which improves its adaptability and reduces the burden of algorithm tuning (Cheng & Prayogo, 2014; Aydoğdu et al., 2016).

Despite their effectiveness, both PSO and SOS encounter substantial computational demands when thousands of design alternatives must be evaluated through finite element analysis (FEA) software such as ETABS or SAP2000. This challenge motivates the integration of dimensionality reduction and machine learning methods to reduce runtime while preserving the accuracy of optimization outcomes.

2.2. Dimensionality Reduction Techniques

DR has become a vital strategy for managing the computational demands of high-dimensional engineering problems. Classical linear approaches, such as PCA, transform correlated variables into a reduced set of independent components while retaining most of the original variance (Pearson, 1901). PCA has found extensive use in areas like structural health monitoring, modal analysis, and feature extraction (Yan et al., 2005).

However, many structural response datasets are nonlinear and highly complex, making linear methods insufficient for capturing critical patterns. To address this, nonlinear DR techniques such as t-SNE and UMAP have been utilized in applications like visualization and clustering (Van der Maaten & Hinton, 2008; McInnes et al., 2018). While these methods preserve structural relationships in data, they are rarely applied in optimization workflows due to computational intensity and challenges with invertibility.

More recently, Autoencoders—a deep learning–based class of DR models—have shown promise. These neural networks compress input data into a latent space and then reconstruct it, capturing nonlinear interactions with minimal reconstruction error (Hinton & Salakhutdinov, 2006). Applications of Autoencoders have included surrogate modeling, system identification, and damage detection (Zhang et al., 2023). Despite their advantages, their integration into real-time optimization remains relatively underexplored.

2.3. Classifier Methods for Structural Feature Spaces

Classification models are increasingly applied to evaluate the quality of reduced-dimensional spaces by assessing the separation between feasible and infeasible designs. Gradient-boosting classifiers such as XGBoost and LightGBM are widely used because of their strong accuracy, computational efficiency, and effectiveness in handling imbalanced data (Chen & Guestrin, 2016; Ke et al., 2017).

In structural optimization, classifiers are trained on labeled datasets where feasibility is defined by code compliance or failure criteria. These models can then predict feasibility in latent space, providing an indirect way to assess whether DR retains essential structural behavior. Importantly, classifiers are not used to guide optimization directly but serve as diagnostic tools to ensure that solutions in reduced space remain valid when decoded back into the original design domain.

3. METHODOLOGY

This paper proposes a hybrid four-stage optimization framework that integrates sample generation, dimensionality reduction, classifier evaluation, and metaheuristic re-optimization to solve large-scale structural design problems more efficiently. In the first stage, a diverse dataset is generated in the original high-dimensional space using metaheuristics such as SOS or PSO. In the second stage, this data is transformed into a reduced-dimensional space using DR techniques, followed by classification using models such as XGBoost, LightGBM, and CatBoost to evaluate the quality of the reduced space in separating feasible and infeasible designs. In the final stage, optimization is rerun in the reduced space using SOS, and solutions are decoded back to the full-dimensional space for validation via ETABS analysis. The full proposed framework can be seen in Fig. 1.

Unlike traditional models that operate solely in high-dimensional domains, the proposed approach leverages DR to reduce computational time, maintain solution accuracy, and improve optimization scalability. The framework is validated using a practical 20-story steel frame case study, demonstrating significant improvements in efficiency without compromising code compliance.

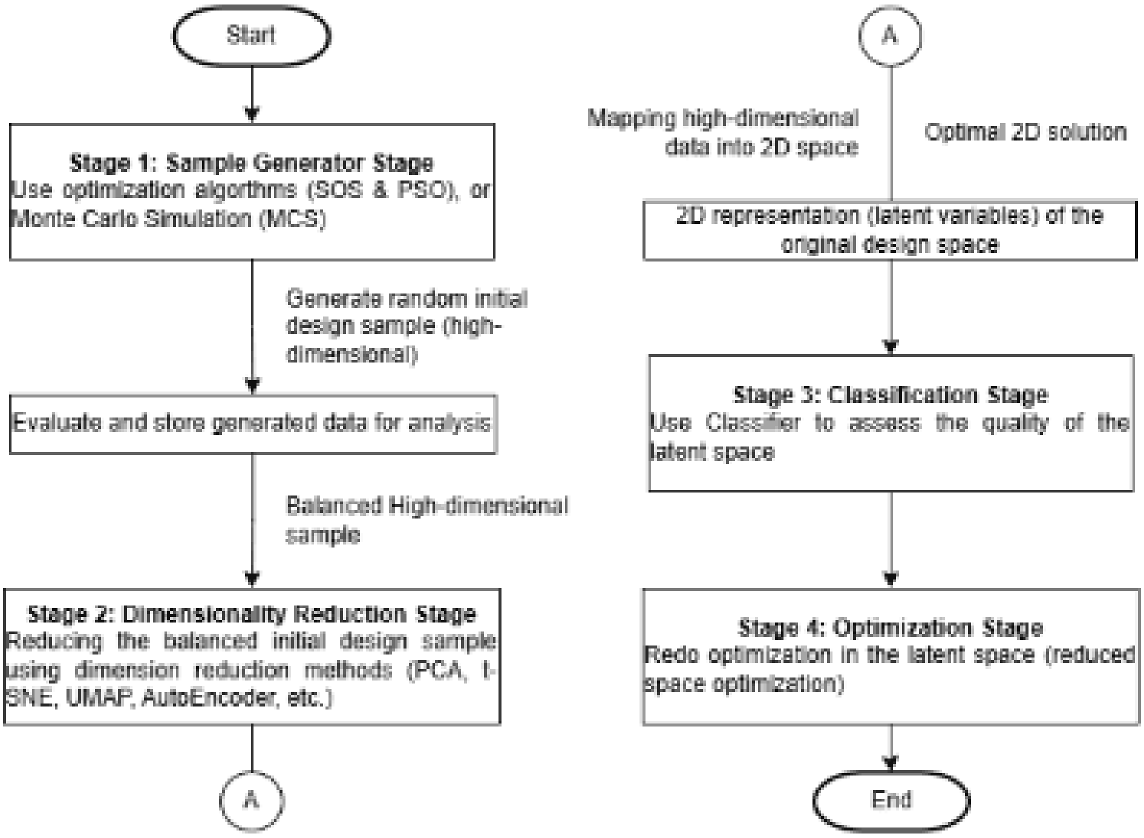

Fig. 1. Proposed Framework

3.1. Stage 1: Sample Generation in High-Dimensional Space

In the first stage, a diverse dataset is generated in the original high-dimensional design space using metaheuristic algorithms such as PSO and SOS. Each candidate design is encoded as a vector of section IDs assigned to structural members (beams, columns, and braces) and then analyzed using structural software—ETABS—via automated API interaction. The objective function typically involves minimizing the total structural weight while satisfying strength and serviceability constraints defined by relevant codes. Each evaluated design is labeled as feasible or infeasible based on its compliance with structural performance criteria. The resulting dataset forms the foundation for the dimensionality reduction process. The optimization objective is to minimize the total structural weight, defined in Equation (1).

$$W = \sum_{i=1}^{n} (\rho_i A_i L_i) \times 2^{\gamma_j} \quad (1)$$

where W is the total weight of the structure, ρ_i is the material density, A_i is the cross-sectional area of member i, L_i is its length, and γ_j represents the penalty for violating constraint j. The term γ_j is dynamic—it increases based on the severity and type of constraint violation, discouraging infeasible solutions during the search. Only designs that pass all strength, stability, and serviceability checks under AISC 360-22 LRFD design provisions are considered feasible. This approach allows for efficient exploration of large design spaces by reducing dimensionality before optimization, enabling the discovery of structurally valid and lightweight solutions with significantly reduced computational cost.

3.2. Stage 2: Dimensionality Reduction

The second stage applies DR techniques to transform the high-dimensional dataset into a compressed latent space, typically 2D for a better visual representation, as well as 5D and 50D to evaluate performance at different levels of dimensionality reduction. Techniques such as PCA, t-SNE, UMAP, and deep-learning-based Autoencoder are utilized. The goal of this stage is to reduce the dimensionality while preserving the structural integrity and design characteristics encoded in the original space. This transformation enables faster search operations and facilitates latent-space diagnostics without relying on full-scale structural re-evaluation.

Autoencoders represent a family of neural networks trained to reconstruct their inputs by learning a compressed representation through a bottleneck layer. They consist of an encoder that maps high-dimensional inputs to a latent space and a decoder that reconstructs the input from the latent representation. Autoencoders are trained using two types of loss functions: Mean Squared Error (MSE) and Mean Absolute Error (MAE), defined respectively as can be seen in Equations (2) and (3).

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2 \tag{2}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|x_i - \hat{x}_i| \tag{3}$$

where x_i and $\hat{x}_i$ denote the original and reconstructed features of the i^{th} dimension, and n is the total number of features in the input vector. These loss functions guide the model to prioritize either large squared deviations (MSE) or uniformly penalize all errors (MAE), influencing the fidelity and generalization capacity of the encoder-decoder structure. Autoencoders with 2, 3, and 5 hidden layers are trained to evaluate the effect of model depth on compression quality and optimization success. Fig. 2a illustrates the Autoencoder architecture used in this study.

This study employs two types of Autoencoder: MAE-based and MSE-based models, each with three different architectural depths: 2-layer (2L), 3-layer (3L), and 5-layer (5L) configurations. It is important to note that all models without the “-V” notation are constructed with a 2-dimensional latent space, used primarily for classification and visualization purposes. The MAE-3L model, for example, consists of a 3-layer encoder and a corresponding 3-layer decoder, optimized using the mean absolute error loss function and has 2 variables in the latent space. Fig. 2b illustrates a 2D projection of the feasible and infeasible areas in the latent space.

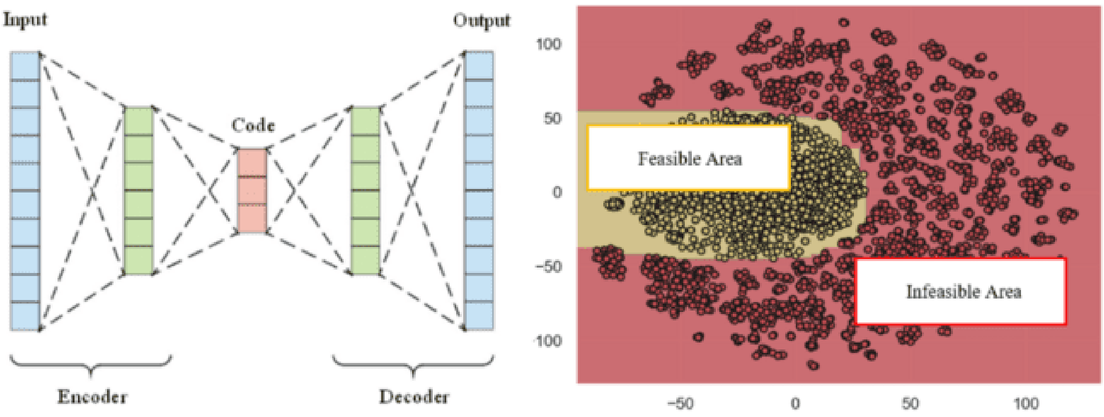

Fig. 2a. Autoencoders Architecture b. Feasible and Infeasible Area in Latent Space.

3.3. Stage 3: Latent Space Diagnosis Using Classifiers

In the third stage, classifier models are applied as diagnostic tools to evaluate the quality of the latent space generated by different DR methods. Gradient boosting models such as XGBoost, LightGBM, and CatBoost are trained on the reduced data, using the original feasibility labels. These classifiers do not participate in the optimization process. Instead, their performance serves as an indirect measure of the separability between “feasible” and “infeasible” regions in the reduced space based on its ability to satisfy all strength and serviceability criteria under AISC LRFD provisions. A strong classifier performance—characterized by high accuracy, precision, recall, F1 score, and Area Under the Curve (AUC), together with competitive runtime—indicates that the DR method successfully preserves critical structural decision boundaries, making it suitable for use in the next stage.

3.4. Stage 4: Re-Optimization in Latent Space and Decoding

In the final stage, the best-performing DR method—validated through classifier diagnostics—is used to guide a re-optimization process in the latent space. The re-optimization is performed using SOS and PSO. Once optimal solutions are found in the latent space, they are decoded back to the original high-dimensional space using inverse transformation (for PCA, UMAP) or decoder networks (for Autoencoder). These decoded solutions are re-analyzed in ETABS to verify structural feasibility and code compliance under multiple LRFD load combinations. This approach effectively narrows the search space, reduces computational effort, and maintains solution fidelity by validating designs both in reduced and full-dimensional domains.

4. RESULTS AND DISCUSSIONS

4.1. Study Case 1: Benchmark Problems

To rigorously benchmark the proposed hybrid optimization framework, eight widely recognized mathematical test functions—Rastrigin, Sphere, Rosenbrock, Ackley, Griewank, Levy, Schwefel, and Zakharov—were employed, as they are commonly used in global optimization research for their diverse topological characteristics such as multimodality, non-convexity, separability, and deceptive local optima. Each function was defined in 100 dimensions, representing 100 decision variables, and subsequently reduced to a two-dimensional latent space using dimensionality reduction techniques. For benchmark testing, Autoencoders with two hidden layers were trained under both MSE and MAE loss functions, providing a balance between training simplicity and the ability to capture meaningful structural patterns in compressed representations. This setup enabled systematic evaluation of the framework's ability to handle high-dimensional problems while retaining essential problem characteristics in a reduced search space.

The comparative evaluation of dimensionality reduction and classification methods is summarized in Table 1, which consolidates accuracy, precision, recall, F1-score, AUC, runtime, and overall rankings. The results show that some algorithm–reduction pairings significantly outperform others; for example, XGBoost achieved the best performance when combined with Factor Analysis (FA), while LightGBM performed strongly with MAE-2L and MSE-2L, both consistently securing top rankings. In contrast, AdaBoost exhibited poor performance across nearly all techniques, particularly with PCA, t-SNE, and UMAP, where it ranked among the lowest. Overall, Table 1 highlights the compatibility of different dimensionality reduction and classification methods, providing practical insights into which combinations deliver robust predictive accuracy and efficiency for high-dimensional structural optimization problems.

Fig. 3 shows the convergence behavior of different dimensionality reduction (DR) techniques on all of the benchmark functions, where the objective is to minimize the function with fewer evaluations. The red colored text indicates the best overall result. The results indicate that most DR methods accelerate convergence compared with the no-DR baseline. PCA and t-SNE achieve the fastest reduction toward low objective values across all functions. Deep Autoencoders such as MSE-2L and MAE-2L also show strong early-stage performance, whereas FA and UMAP converge more slowly and with less stability.

Table 1. Dimensionality Reduction – Classifier Result for Benchmark Problems. (The red colored text indicates the best overall result for each metric across all dimensionality reduction methods)

Testing Accuracy	PCA	t-SNE	UMAP	FA	MAE-2L	MSE-2L
XGBoost	0.836	0.835	0.942	**0.976**	0.982	0.975
LightGBM	0.899	0.828	0.935	0.975	0.981	0.974
CatBoost	0.899	0.833	0.943	0.978	0.984	0.978
AdaBoost	0.899	0.824	0.853	0.943	0.911	0.938

Testing Precision	PCA	t-SNE	UMAP	FA	MAE-2L	MSE-2L
XGBoost	0.869	0.809	0.952	**0.977**	0.982	0.976
LightGBM	0.900	0.809	0.956	0.980	0.984	0.980
CatBoost	0.900	0.807	0.957	0.979	0.984	0.979
AdaBoost	0.915	0.835	0.902	0.950	0.925	0.946

Testing Recall	PCA	t-SNE	UMAP	FA	MAE-2L	MSE-2L
XGBoost	0.836	0.835	0.942	**0.976**	0.982	0.975
LightGBM	0.899	0.828	0.935	0.975	0.981	0.974
CatBoost	0.899	0.833	0.943	0.978	0.984	0.978
AdaBoost	0.899	0.824	0.853	0.943	0.911	0.938

Testing F1 Score	PCA	t-SNE	UMAP	FA	MAE-2L	MSE-2L
XGBoost	0.769	0.821	0.944	**0.976**	0.982	0.975
LightGBM	0.853	0.817	0.941	0.976	0.982	0.976
CatBoost	0.853	0.818	0.946	0.978	0.984	0.979
AdaBoost	0.853	0.772	0.825	0.918	0.872	0.914

AUC	PCA	t-SNE	UMAP	FA	MAE-2L	MSE-2L
XGBoost	0.549	0.612	0.962	**0.991**	0.994	0.990
LightGBM	0.503	0.532	0.962	0.991	0.994	0.990
CatBoost	0.664	0.603	0.966	0.992	0.995	0.992
AdaBoost	0.525	0.560	0.903	0.979	0.915	0.982

Runtime (s)	PCA	t-SNE	UMAP	FA	MAE-2L	MSE-2L
XGBoost	1.460	0.138	0.146	**0.140**	1.552	0.150
LightGBM	1.235	0.620	0.614	0.585	1.107	0.619
CatBoost	12.964	13.960	12.915	14.023	16.581	14.842
AdaBoost	3.478	4.205	4.472	3.331	3.376	3.870

Rank	PCA	t-SNE	UMAP	FA	MAE-2L	MSE-2L
XGBoost	16	17	7	**1**	5	3
LightGBM	15	19	8	6	2	4
CatBoost	18	22	12	9	11	10
AdaBoost	21	24	23	13	20	14

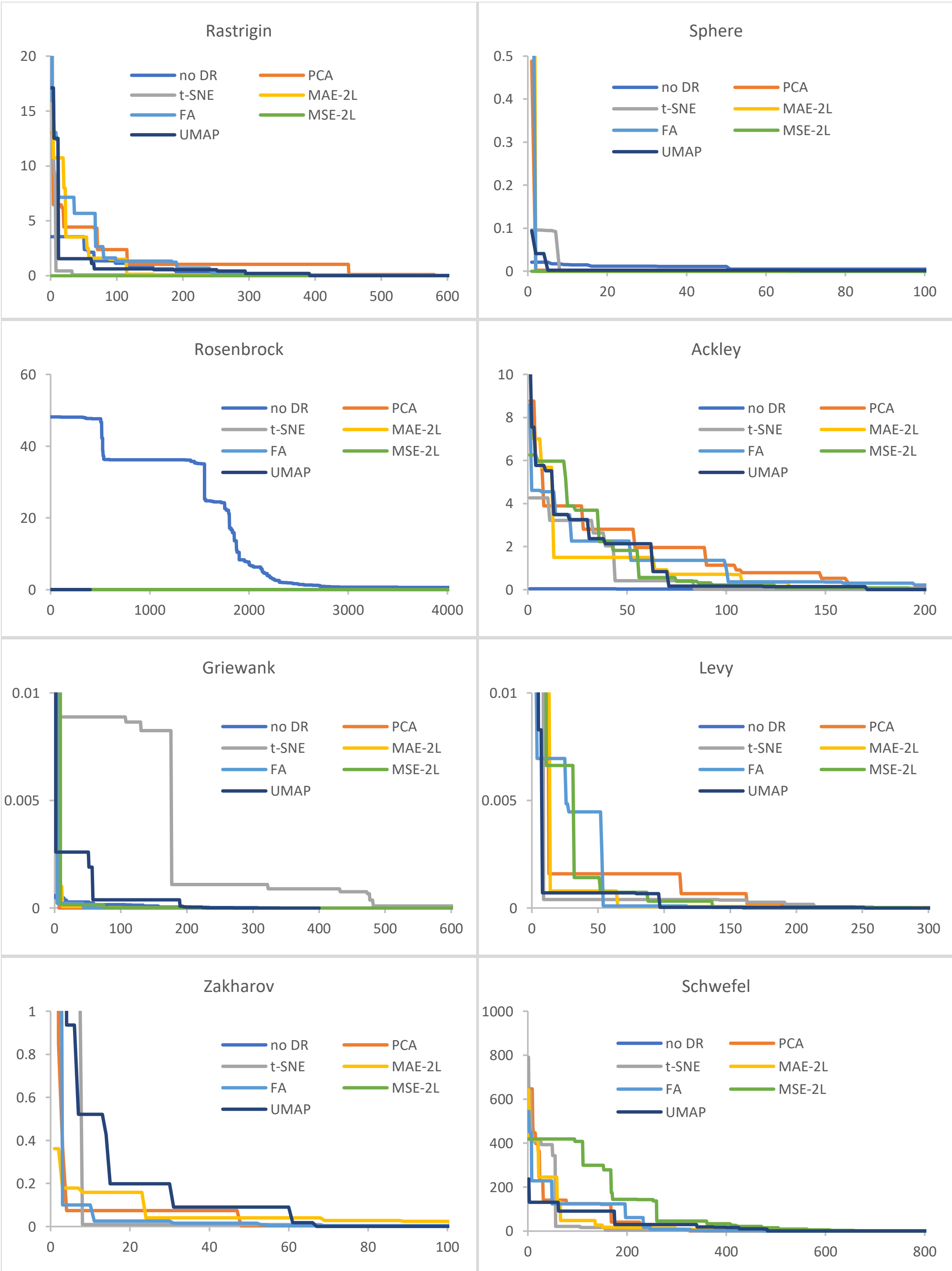

Fig. 3. Convergence curve of the first optimization, where the x-axis represents the number of evaluations and the y-axis represents the fitness value.

Table 2 compares the performance of selected dimensionality reduction (DR) methods across eight benchmark functions. PCA and t-SNE consistently delivered the best balance of fitness accuracy and runtime, often converging within less than one second and achieving near-optimal values on functions such as Rastrigin, Sphere, and Rosenbrock. Factor Analysis (FA) also performed competitively, providing stable convergence with low computational demand. Autoencoders, specifically MSE-2L and MAE-2L, ranked just below PCA and t-SNE, demonstrating strong accuracy and robustness while requiring slightly higher runtimes. In contrast, UMAP repeatedly showed severe inefficiency, taking hundreds of seconds on most functions despite producing fitness values comparable to other methods.

Table 2. Summary of the Second Optimization for Benchmark Problems

Best Fitness	no DR	PCA	t-SNE	UMAP	FA	MSE-2L	MAE-2L
Rastrigin	6.503E-07	7.571E-07	2.345E-07	6.816E-07	3.928E-07	5.985E-07	1.049E-05
Sphere	8.707E-07	4.854E-07	2.017E-07	3.551E-07	2.264E-07	0.000E+00	0.000E+00
Rosenbrock	6.639E-02	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00	0.000E+00
Ackley	9.105E-07	2.645E-07	5.101E-07	3.896E-07	7.792E-07	9.538E-07	9.721E-07
Griewank	7.561E-07	7.850E-07	7.391E-07	2.025E-07	1.340E-07	2.845E-07	4.293E-07
Levy	8.754E-07	5.444E-07	8.405E-09	2.199E-07	1.206E-07	8.785E-08	8.629E-07
Schwefel	5.776E+01	2.546E-05	2.546E-05	2.546E-05	2.546E-05	2.546E-05	2.546E-05
Zakharov	9.806E-07	6.478E-07	6.008E-08	1.185E-07	6.486E-09	2.320E-08	8.933E-07

Convergence (Iteration)	no DR	PCA	t-SNE	UMAP	FA	MSE-2L	MAE-2L
Rastrigin	29	26	29	21	22	22	19
Sphere	18	7	2	5	6	1	5
Rosenbrock	1000	1	1	1	1	1	1
Ackley	39	20	18	19	19	19	19
Griewank	12	1	22	7	2	6	6
Levy	567	12	11	9	9	6	10
Schwefel	1000	11	14	11	10	9	10
Zakharov	355	8	9	10	8	8	6

Time (s)	no DR	PCA	t-SNE	UMAP	FA	MSE-2L	MAE-2L
Rastrigin	5.732	0.870	1.053	921.483	1.168	1.093	4.663
Sphere	2.691	0.233	0.100	367.107	0.236	0.112	0.633
Rosenbrock	200.579	1.421	0.121	2051.498	0.086	0.080	0.473
Ackley	4.815	0.387	0.503	291.274	0.904	0.626	1.827
Griewank	2.495	0.069	0.615	280.734	0.267	0.239	0.691
Levy	70.653	0.274	0.259	484.688	0.325	0.777	0.722
Schwefel	219.165	0.315	0.496	11.676	0.527	0.408	0.728
Zakharov	37.970	0.210	0.206	143.277	0.392	0.272	0.288

The benchmark results represent the third stage of the proposed optimization framework, where the optimizer operates within the dimensionality-reduced latent space constructed by each DR method. The "no DR" column serves as the baseline, representing direct optimization in the original 20-dimensional design space without any dimensionality reduction. By comparing each DR method against this baseline, the table demonstrates how effectively each technique preserves optimization landscape characteristics while reducing computational complexity. The superior performance of PCA and t-SNE in both convergence speed (iteration count) and computational time validates their ability to construct latent spaces that retain the essential features needed for efficient optimization, while UMAP's extended runtimes suggest its manifold learning approach introduces unnecessary computational overhead for this structural optimization problem.

4.2. Study Case 2: Structural Problem

The structural case study involves the design optimization of a 20-story braced steel space frame composed of 960 elements shown in Fig. 4, each treated as an independent design variable corresponding to a discrete steel section from the *Steel Construction Manual* published by the American Institute of Steel Construction (2017). The model is based on the benchmark configuration proposed by Aydoğdu, Saka, and Hasançebi (2016). The total building height is 73.2 meters, with a typical floor height of 3.66 meters and a regular rectilinear grid of beams, columns, and concentric bracing. All supports are fixed in both translation and rotation to simulate rigid foundation conditions.

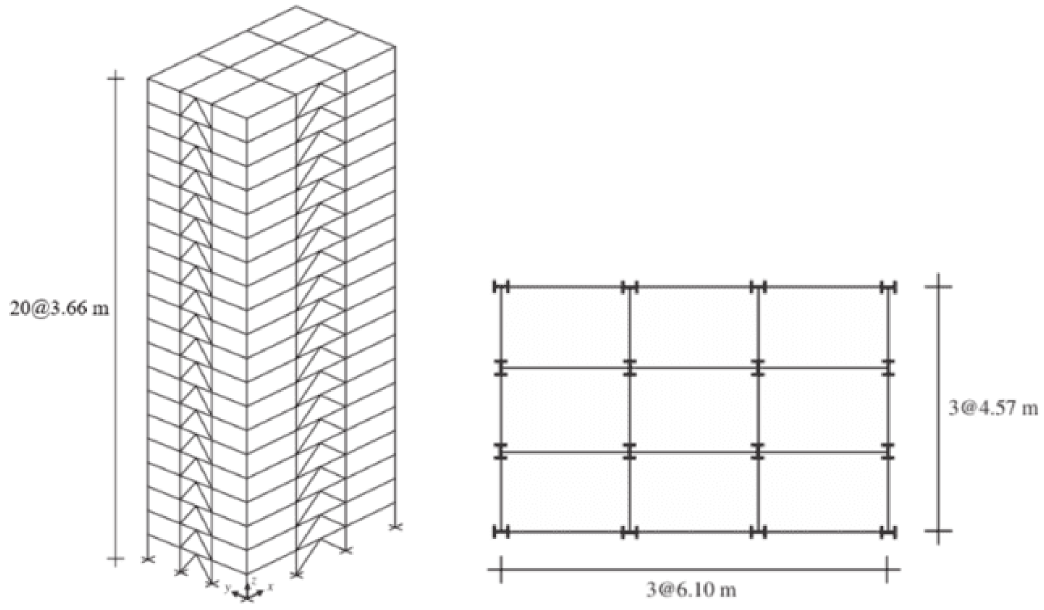

Fig. 4. 3D and plan view of the 20-story 3D steel frame structure.

Material properties assumed in the analysis include a modulus of elasticity of 200 GPa and a yield stress of 250 MPa for structural steel. The structure is subjected to gravity, snow, and wind loads, modeled according to ASCE 7-16. Floor levels 1–19 carry a dead load of 2.88 kN/m² and a live load of 2.39 kN/m², while the roof includes an additional snow load of 1.20 kN/m². Wind loads are based on a basic wind speed of 46.94 m/s, with exposure category B and standard wind coefficients. Two lateral loading conditions are applied: wind in the *x*-direction and *y*-direction, both combined with gravity.

In the first-stage optimization, the objective was to generate a minimum of 25 feasible and code-compliant structural designs, which would serve as the training set for the subsequent dimensionality reduction and classification stages. Additionally, the column sections were arranged such that their sizes increase progressively from the top stories to the bottom, ensuring practical constructability and structural stability. Rather than seeking a global optimum, the optimization process was terminated as soon as this requirement was fulfilled. To ensure a fair comparison, both SOS and PSO were initialized with the same random seed, allowing for consistent sampling conditions and unbiased performance evaluation. Among the two metaheuristic algorithms tested, SOS proved to be significantly more effective in both convergence speed and solution quality. It required only 303 evaluations to generate 25 feasible designs, while PSO needed 910 evaluations to achieve the same, shown in Table 3.

Table 3. Summary of the First Optimization

Algorithm	Enough sample at	min fitness (kg)	time (s)
SOS	303	776393.580	3741.745
PSO	910	808206.533	11156.272

Moreover, the best fitness value (lowest structural weight) obtained by SOS was 776393.580 kg, compared to 808206.533 kg produced by PSO. This indicates that SOS is not only faster but also better at identifying lighter, code-compliant configurations early in the search process. In terms of computational time, SOS completed the sampling phase in 3741.745 seconds—substantially faster than PSO's 11156.272 seconds. The SOS optimization process was stopped after 303 evaluations, as it had already generated the required feasible samples for the next stage. In contrast, PSO continued until 910 evaluations to reach the same threshold. Based on this superior performance in both efficiency and effectiveness, only the SOS-generated dataset was selected for the remaining stages of this study, including dimensionality reduction, classification, and the final second-stage optimization. SOS is also retained as the optimization engine for the reduced space search. Fig. 5 shows the convergence behavior of both algorithms during the first-stage optimization.

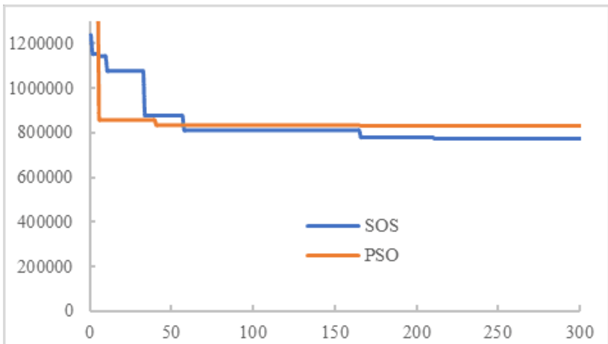

Fig. 5. Convergence curve of the first optimization, where the x-axis represents the number of evaluations and the y-axis represents the fitness value in kg.

Table 4 shows that among all dimensionality reduction and classifier combinations tested, the integration of a 5-layer Autoencoder trained with MSE and 5 variables in the latent space (MSE-5L-5V) with XGBoost consistently outperforms others across nearly all evaluation metrics, including accuracy, precision, recall, F1-score, and AUC. The strong classification results indicate that MSE-5L-5V effectively captures essential structural characteristics in its latent representation, enabling XGBoost to distinguish feasible from infeasible solutions with high reliability. While the MSE-5L-50V model achieves the highest absolute scores in several metrics, its runtime is almost 20 times slower than MSE-5L-5V, making it impractical for large-scale applications. In contrast, traditional methods such as PCA, though computationally faster, exhibited weaker classification performance, while nonlinear manifold techniques like t-SNE and UMAP struggled to preserve global separability required for accurate feasibility prediction. Overall, Table 4 confirms that the MSE-5L-5V-XGBoost combination provides the most effective balance between classification accuracy and computational efficiency, establishing it as the optimal configuration for evaluating latent space quality within the proposed framework. To clarify the metrics; accuracy, precision, recall, F1-score, and AUC are performance ratios from 0 to 1, where higher values indicate better classification of feasible versus infeasible designs. Runtime is reported in seconds, and rank represents overall performance ordering (1 = best). The consistently high values (>0.99) achieved by MSE-5L-5V demonstrate that this autoencoder preserves the critical constraint boundaries of the 20-dimensional structural problem within a compact 5-dimensional latent space, enabling accurate feasibility classification with minimal computational cost.

Table 4. Dimensionality Reduction – Classifier Result for Structural Problem

Testing Accuracy	PCA	t-SNE	UMAP	FA	MAE-2L	MAE-3L	MAE-5L	MSE-2L	MSE-3L	MSE-5L	MSE-5L-5V	MSE-5L-50V
XGBoost	0.913	0.870	0.913	0.891	0.902	0.924	0.902	0.902	0.902	0.902	**0.997**	0.999
LightGBM	0.924	0.913	0.913	0.913	0.880	0.967	0.924	0.891	0.935	0.891	0.994	0.999
CatBoost	0.902	0.880	0.891	0.891	0.913	0.935	0.891	0.913	0.935	0.891	0.989	0.999
AdaBoost	0.913	0.913	0.913	0.913	0.913	0.913	0.913	0.913	0.913	0.913	0.987	0.999

Testing Precision	PCA	t-SNE	UMAP	FA	MAE-2L	MAE-3L	MAE-5L	MSE-2L	MSE-3L	MSE-5L	MSE-5L-5V	MSE-5L-50V
XGBoost	0.894	0.896	0.886	0.891	0.870	0.920	0.885	0.885	0.870	0.885	**0.997**	0.999
LightGBM	0.920	0.913	0.903	0.894	0.873	0.966	0.920	0.903	0.929	0.878	0.994	0.998
CatBoost	0.897	0.887	0.903	0.891	0.886	0.939	0.862	0.913	0.926	0.862	0.989	0.999
AdaBoost	0.921	0.921	0.921	0.921	0.921	0.921	0.921	0.921	0.921	0.921	0.987	0.999

Testing Recall	PCA	t-SNE	UMAP	FA	MAE-2L	MAE-3L	MAE-5L	MSE-2L	MSE-3L	MSE-5L	MSE-5L-5V	MSE-5L-50V
XGBoost	0.913	0.870	0.913	0.891	0.902	0.924	0.902	0.902	0.902	0.902	**0.997**	0.999
LightGBM	0.924	0.913	0.913	0.913	0.880	0.967	0.924	0.891	0.935	0.891	0.994	0.999
CatBoost	0.902	0.880	0.891	0.891	0.913	0.935	0.891	0.913	0.935	0.891	0.989	0.999
AdaBoost	0.913	0.913	0.913	0.913	0.913	0.913	0.913	0.913	0.913	0.913	0.987	0.999

Testing F1 Score	PCA	t-SNE	UMAP	FA	MAE-2L	MAE-3L	MAE-5L	MSE-2L	MSE-3L	MSE-5L	MSE-5L-5V	MSE-5L-50V
XGBoost	0.900	0.881	0.888	0.891	0.881	0.922	0.892	0.892	0.881	0.892	**0.997**	0.998
LightGBM	0.922	0.913	0.907	0.900	0.877	0.966	0.922	0.897	0.931	0.884	0.994	0.998
CatBoost	0.899	0.884	0.897	0.891	0.888	0.916	0.874	0.913	0.925	0.874	0.986	0.998
AdaBoost	0.872	0.872	0.872	0.872	0.872	0.872	0.872	0.872	0.872	0.872	0.981	0.998

AUC	PCA	t-SNE	UMAP	FA	MAE-2L	MAE-3L	MAE-5L	MSE-2L	MSE-3L	MSE-5L	MSE-5L-5V	MSE-5L-50V
XGBoost	0.958	0.920	0.938	0.928	0.926	0.966	0.943	0.909	0.921	0.929	**0.991**	0.941
LightGBM	0.945	0.847	0.798	0.876	0.884	0.970	0.945	0.827	0.958	0.921	0.987	0.864
CatBoost	0.938	0.926	0.890	0.923	0.914	0.954	0.921	0.945	0.958	0.915	0.991	0.947
AdaBoost	0.935	0.946	0.833	0.935	0.940	0.938	0.935	0.908	0.912	0.945	0.631	0.852

Runtime (s)	PCA	t-SNE	UMAP	FA	MAE-2L	MAE-3L	MAE-5L	MSE-2L	MSE-3L	MSE-5L	MSE-5L-5V	MSE-5L-50V
XGBoost	0.051	0.037	0.035	0.049	0.045	0.034	0.035	0.043	0.036	0.036	**0.109**	1.739
LightGBM	0.057	0.050	0.046	0.058	0.050	0.052	0.059	0.109	0.053	0.050	0.201	1.615
CatBoost	3.959	4.056	4.134	4.101	10.886	9.989	10.304	10.675	11.078	10.887	11.307	65.104
AdaBoost	0.751	0.696	0.702	0.705	0.846	0.791	0.808	0.809	0.859	0.802	4.905	49.471

Rank	PCA	t-SNE	UMAP	FA	MAE-2L	MAE-3L	MAE-5L	MSE-2L	MSE-3L	MSE-5L	MSE-5L-5V	MSE-5L-50V
XGBoost	18	45	32	39	41	13	34	36	42	35	**1**	3
LightGBM	14	17	19	23	48	6	15	38	10	43	2	5
CatBoost	31	44	37	40	33	11	46	16	12	47	4	8
AdaBoost	26	20	30	25	22	24	27	29	28	21	9	7

The re-optimization stage was conducted using SOS across all dimensionality reduction methods. To draw a comparison, the no-DR (no dimensionality reduction) approach is the baseline, where only SOS is applied directly to the original high-dimensional input space without any transformation or compression, as shown in Fig. 6 and summarized in Table 5. The best performance was achieved by the 5-layer MSE-trained autoencoder (MSE-5L), which reached a fitness value of 691498.065 kg — outperforming all other configurations, including the no-DR baseline. It also converged in only 4978 evaluations, significantly faster than the 5450 evaluations required by the no-DR approach.

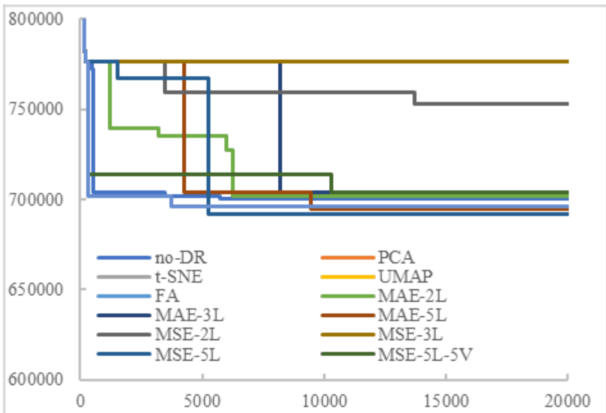

Fig. 6. Convergence curve of the second optimization, where the x-axis represents the number of evaluations and the y-axis represents the fitness value in kg.

Table 5. Summary of the Second Optimization for Structural Problem

Dimensionality Reducer	Convergence (Iterations)	fitness (kg)	time (to max FE, s)
no-DR	5450	700002.672	258434.725
PCA	1	776393.580	371063.676
t-SNE	1	776393.580	542401.910
UMAP	1	776393.580	625712.752
FA	1	776393.580	670320.363
Autoencoder MAE-2L	5967	701993.688	441054.540
Autoencoder MAE-3L	7897	704190.554	610394.874
Autoencoder MAE-5L	9150	694339.474	287048.621
Autoencoder MSE-2L	19705	704190.554	301329.737
Autoencoder MSE-3L	1	776393.580	316629.658
Autoencoder MSE-5L	4978	691498.065	379891.162
Autoencoder MSE-5L-5V	2125	695815.431	391631.804
Autoencoder MSE-5L-50V	3447	691632.107	737042.763

Shallow autoencoders and linear methods like PCA produced similar results (fitness = 776393.580), but t-SNE, UMAP, and FA's last improvement occurred in the first iteration, suggesting inaccurate prediction and poor search diversity. While the no-DR case yielded good performance, it had a slightly higher number of convergence iterations, reaching 5450. The MSE-5L model produced better designs, achieving lower objective function values compared to the no-DR case. These results confirm that dimensionality reduction can enhance solution quality in structural optimization. However, using the 5V and 50V variants may yield slightly better designs, but they require significantly longer computation times, indicating a clear time–cost trade-off that must be considered in practical applications.

5. CONCLUSIONS

This study introduced a hybrid optimization framework that integrates DR techniques with metaheuristic algorithms to address high-dimensional structural design problems. By combining methods such as PCA, t-SNE, UMAP, and deep Autoencoders with SOS and PSO, the framework reduces computational burden while maintaining solution quality. Applied to benchmark functions and the optimization of a 20-story braced steel frame, the results showed that deep Autoencoders—particularly MSE-trained models—captured complex structural patterns more effectively than shallow projection-based techniques. These models achieved lower structural weight and faster convergence compared to the no-DR baseline, while SOS consistently outperformed PSO in generating feasible and code-compliant designs. Overall, PCA and t-SNE emerged as the most practical techniques, while FA and Autoencoders offer reliable alternatives when nonlinear structural information needs to be preserved.

Key contributions of this study include: (1) demonstrating that DR can reduce computational effort without sacrificing design accuracy; (2) validating SOS as a robust optimizer for high-dimensional structural problems; and (3) highlighting the importance of Autoencoder depth and loss function selection, with MSE-based models achieving the best balance of accuracy and efficiency. The framework also integrates classifier-based diagnostics to ensure feasibility in the reduced design space and incorporates constructability constraints, bridging computational optimization with practical engineering requirements.

Future research will extend this framework to larger-scale and more heterogeneous structural systems, explore advanced Autoencoder architectures with adaptive loss functions, and investigate uncertainty modeling and surrogate-based optimization to further improve scalability and robustness in real-world applications. A key limitation of the present study is that, unlike practical steel structure design, where grouping by two or three floors is typically applied to reduce design variables, this study did not adopt grouping. This decision was intentional, as the objective was to directly evaluate the true impact of dimensionality reduction on the full set of variables without masking effects from pre-grouping. However, this assumption makes the case study less representative of real-world engineering practice. Future research should therefore address larger and more realistic structures with thousands of members, where grouping strategies can be integrated with dimensionality reduction to compress the design space from thousands of variables to a few hundred, while still maintaining computational tractability and engineering relevance.

6. REFERENCES

American Institute of Steel Construction (AISC). (2022). *Specification for Structural Steel Buildings*. Chicago, IL: AISC.

Aydoğdu, İ., Saka, M. P., & Hasançebi, O. (2016). Parameter-free metaheuristics for steel design. Advances in Engineering Software, 92, 1–14.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 785–794). San Francisco, CA: ACM. https://doi.org/10.1145/2939672.2939785

Cheng, M. Y., & Prayogo, D. (2014). Symbiotic organisms search: A new metaheuristic optimization algorithm. Computers & Structures, 139, 98–112. https://doi.org/10.1016/j.compstruc.2014.03.007

Gandomi, A. H., Yang, X. S., & Talatahari, S. (2013). Metaheuristic algorithms in structural engineering. London: Springer.

Hasançebi, O., Çarbaş, S., Doğan, E., Erdal, F., & Saka, M. P. (2011). Performance evaluation of metaheuristic search techniques in the optimum design of real size pin jointed structures. Computer-Aided Civil and Infrastructure Engineering, 26(2), 135–147.

Hinton, G. E., & Salakhutdinov, R. R. (2006). Reducing dimensionality with neural networks. Science, 313(5786), 504–507. https://doi.org/10.1126/science.1127647

Kaveh, A., & Mahdavi, V. R. (2014). Optimal design of frames using PSO. Advances in Engineering Software, 76, 82–92.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., & Liu, T. Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. In Advances in Neural Information Processing Systems (NeurIPS), 30 (pp. 3146–3154).

Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In Proceedings of the IEEE International Conference on Neural Networks (ICNN) (pp. 1942–1948). Perth, Australia: IEEE. https://doi.org/10.1109/ICNN.1995.488968

McInnes, L., Healy, J., & Melville, J. (2018). UMAP: Uniform manifold approximation and projection. Journal of Open-Source Software, 3(29), 861. https://doi.org/10.21105/joss.00861

Pearson, K. (1901). On lines and planes of closest fit to systems of points. Philosophical Magazine, 2(11), 559–572. https://doi.org/10.1080/14786440109462720

Van der Maaten, L., & Hinton, G. (2008). Visualizing data using t-SNE. Journal of Machine Learning Research, 9, 2579–2605.

Yan, A. M., Kerschen, G., De Boe, P., & Golinval, J. C. (2005). Structural damage detection using PCA. Mechanical Systems and Signal Processing, 19(2), 341–356.

Zhang, Y., Li, X., Wang, L., & Zhou, Y. (2023). Autoencoder-based surrogate models in structural mechanics. Engineering Structures, 283, 115–232.