Review | Received 25 May 2026; Revised 15 July 2026; Accepted 22 July 2026; Published 6 August 2026
https://doi.org/10.55092/sc20260014

Reinforcement learning for intelligent optimization of steel structures: a review

Yuqing Gao[1], **Yuhang Lu**[1,*], **Meiyu Du**[2], **Wei Wang**[2] **and Guanren Zhou**[3]

[1] Department of Disaster Mitigation for Structures, Tongji University, Shanghai, China
[2] Department of Structural Engineering, Tongji University, Shanghai, China
[3] Department of Civil and Environmental Engineering, University of California, Berkeley, USA

* Corresponding author; E-mail: yuhanglu@tongji.edu.cn.

Highlights:

- Reviews the use of reinforcement learning for section optimization in steel structures.
- Compares traditional, surrogate-assisted, and reinforcement learning-based optimization methods.
- Identifies key challenges and future directions for developing code-compliant, efficient, generalizable, and engineering-oriented reinforcement learning-based optimization methods.

Abstract: The optimization of steel structures encompasses a range of design tasks, including section and topology optimization, as well as the optimization of detailing and device parameters. For steel structures, section optimization is a representative task in which component sections are selected, adjusted, and verified through structural analysis and code-compliance checks. Traditional optimization algorithms have been widely used to identify feasible and economical designs, while surrogate-assisted methods have also been implemented to reduce the cost of repeated structural evaluations. However, these approaches are generally organized around case-specific searches rather than around the learning of reusable decision policies across related structural instances. Reinforcement learning (RL) provides a complementary perspective by learning optimization strategies through interaction with structural environments. This review examines the use of RL for the intelligent optimization of steel structures, focusing on the section optimization of steel frames. First, the scope of the review is defined by organizing the main optimization tasks and forms of design variables in steel structures. Traditional optimization and surrogate-assisted methods are then reviewed as methodological foundations for single-case searches, constraint handling, and accelerated evaluation. Building on these foundations, RL-based studies are synthesized from the perspectives of structural-family formulation, environment-based policy learning, and generalization evaluation. The key barriers to developing code-compliant and engineering-oriented RL methods for steel-structure optimization are also discussed. This review provides a structured synthesis for understanding the role of RL in structural optimization and for developing more generalizable, verifiable, and engineering-oriented intelligent design methods.

Keywords: reinforcement learning; steel structures; steel moment-resisting frames; steel-braced frames; intelligent optimization

1. Introduction

Steel structures have played a pivotal role in modern construction due to their high strength-to-weight ratio, favorable seismic performance, and compatibility with industrialized fabrication and assembly-based construction. The suitability of steel frames for standardized and modular construction enables rapid construction and facilitates quality control, functional retrofitting, and damage repair throughout the service life of a building. This advantage is particularly pronounced in high-rise and super-tall buildings, long-span spatial structures, and projects with stringent seismic design requirements, where steel moment-resisting frames and steel-braced frames are commonly adopted to balance safety, economy, and construction efficiency [1–3].

The structural system and bracing layout of steel structures are usually determined at the conceptual design stage. Once these system-level decisions are fixed, they are difficult to modify substantially, for they are constrained by architectural requirements, design codes, and construction conditions. Under a given structural layout, component-level section optimization therefore becomes one of the most direct and practical ways to improve material efficiency while maintaining structural safety and code compliance. Its effects on steel consumption, component utilization, and overall construction cost are also more readily quantified than those of early-stage scheme adjustments [3–5]. For structures with a large number of components and long spans, such as steel petrochemical pipe racks and transfer layers in airport suspended ceiling systems, even modest improvements in component design may yield substantial material savings and cost reductions at the project scale [6,7].

A central challenge in the optimization of steel structures is the efficient solution of highly constrained problems involving discrete design variables and, in many cases, multiple objectives. Structural safety, detailing provisions, and code requirements are constraints, and the objective is to achieve material efficiency while maintaining adequate safety margins. The complexity of this problem is mainly reflected in three aspects. First, the use of discrete section pools and component grouping leads to an exponential growth in the combinatorial scale. Even a medium-sized frame may correspond to an enormous number of possible section-assignment combinations, resulting in a high-dimensional and discontinuous search space [5,8]. Second, structural designs must be evaluated under multiple load cases and against multiple performance metrics. System-level responses (e.g., interstory drift ratio and floor acceleration) are coupled with component-level checks (e.g., stress ratio and stability verification), and unavoidable trade-offs exist among competing objectives and constraints [9,10]. Third, finite element (FE) analysis and code-compliance checks are computationally expensive, particularly for three-dimensional systems or cases involving nonlinear dynamic analysis. The cost of a single evaluation can therefore become substantial, which in turn limits the number of affordable optimization iterations [11–13].

Traditional optimization methods for steel structures have been extensively developed, with metaheuristic algorithms such as the genetic algorithm (GA) [14], particle swarm optimization (PSO) [15], and simulated annealing (SA) [16] forming representative approaches. Surrogate models have also been widely introduced to reduce the computational cost of repeated structural analysis and design evaluation. Existing studies mainly use such models to approximate key structural responses, such as drift and acceleration [2,3], or to predict whether candidate designs satisfy major constraints [17,18]. Surrogate models have also been extended to design-evaluation metrics, such as recovery time, to support recovery-oriented optimization [19].

However, these methods are usually developed and implemented for individual design cases, and the search process is often restarted for each new project. As a result, the optimization experience or the search trajectory accumulated in one case is difficult to transfer to related structural instances. These limitations motivate a paradigm shift toward algorithms capable of learning reusable optimization strategies that generalize across structural instances, thereby effectively reducing optimization time.

With the rapid development of artificial intelligence (AI), RL has emerged as a promising approach for overcoming the bottlenecks of traditional optimization methods in both computational efficiency and experience reuse. Since the breakthrough of AlphaGo in 2016 [20], RL has achieved remarkable success in a wide range of domain-specific applications, including complex games [21], robotic control [22,23], autonomous driving [24], and magnetic control of tokamak plasmas [25]. The development of RL algorithms has also progressed beyond widely used model-free approaches to include model-based algorithms such as Dreamer [26]. In addition, RL has recently been used as an important post-training technique for enhancing the reasoning ability of large language models, as exemplified by DeepSeek-R1 [27].

RL is commonly formulated as a Markov decision process (MDP), denoted as $M = (S, A, P, R, \gamma)$, where S is the state space, A is the action space, $P(s_{t+1} \mid s_t, a_t)$ is the transition function, $R(s_t, a_t, s_{t+1})$ is the reward function, and γ is the discount factor. At each decision step, the agent observes the current state, selects an action according to its policy, receives a reward, and moves to the next state according to the transition dynamics. For the optimization of steel structures, the current design state can be represented by section assignments, structural responses, and constraint margins, while actions correspond to section modifications or topology adjustments. After each action, finite element analysis and code-compliance checks provide updated feedback, allowing the policy to learn how design modifications influence feasibility and performance [7,28,29]. Compared with conventional case-specific search, RL aims to learn a generalizable adjustment policy from many related structural instances, so that the trained policy can improve the efficiency of solving optimization tasks and, in some cases, the resulting design quality. Nevertheless, the current evidence for such generalization is still developing. Existing RL-based studies have mainly been evaluated within specific structural examples and predefined design settings, while robust transferability to substantially different geometries, loading patterns, section pools, and code-compliance requirements still requires further validation. Therefore, learned policies should be regarded as design-support tools for generating or modifying candidate designs, which must undergo structural analysis and code-compliance verification before engineering use.

Recent studies have also discussed AI-supported structural design workflows and generative design methods for building structures and modular steel structures [30,31]. Building on this broader background, this review focuses on steel structures, especially steel moment-resisting frames and steel-braced frames, with particular emphasis on component section optimization. The review surveys representative studies on traditional optimization and RL approaches, with surrogate-assisted optimization discussed as an important accelerator. Special attention is given to RL-based approaches under stringent code constraints and computationally expensive finite element analyses, including their modeling formulations, reward and training strategies, feasibility assurance, and computational cost control. Existing efforts on transferability and generalization across related structural systems are also examined. The review framework and technical scope of this paper are summarized in Figure 1.

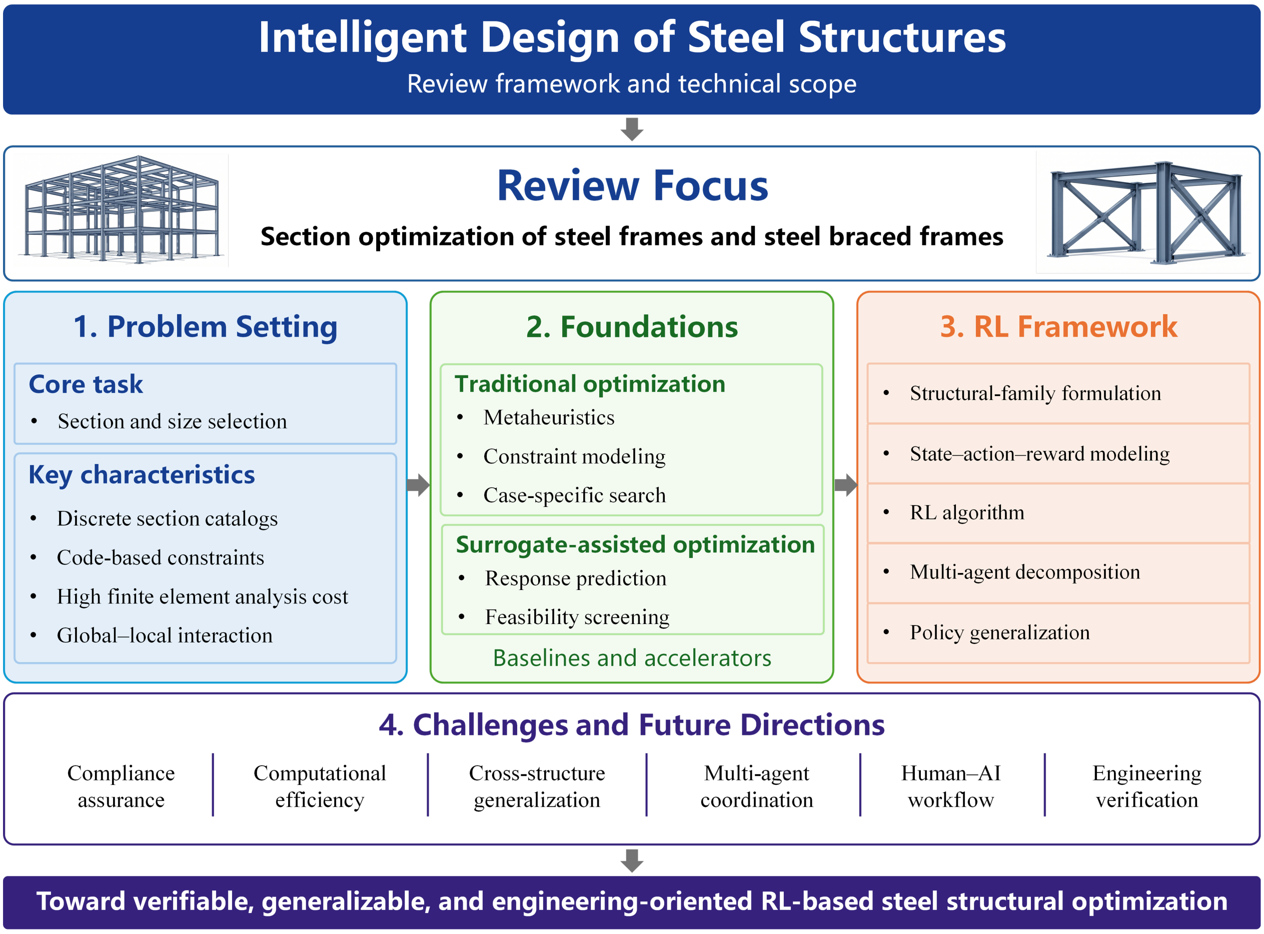

Figure 1. Review framework and technical scope.

2. Analysis of publications

To clarify the research foundation and recent progress of RL-based optimization of steel structures, relevant studies were retrieved, screened, and organized. The literature was mainly retrieved from the Scopus database. The search primarily focused on journal articles written in English, with the document type limited to Article and the subject area limited to Engineering. Considering that the application of RL and AI methods in structural engineering has mainly developed over the past decade, the search mainly covered studies published from 2015 to 2026. Several earlier classical studies on structural optimization were also included through reference tracing and manual supplementation.

The keywords used in this review were divided into two main groups.

(1) The first group was used to identify studies related to RL and structural optimization:

("reinforcement learning") AND ("structural optimization" OR "structural design") AND ("steel frame" OR "steel moment-resisting frames" OR "steel-braced frame" OR "truss" OR "steel structure")

This group of keywords was mainly used to retrieve studies on the application of RL to the optimization of steel moment-resisting frames, steel-braced frames, trusses, and related structural systems.

(2) The second group was used to identify studies on traditional and surrogate-model-assisted optimization methods for steel structures:

("steel frame" OR "steel moment-resisting frames" OR "steel-braced frame" OR "truss" OR "steel structure") AND ("structural component optimization" OR "structure optimization" OR "design optimization") AND ("metaheuristic" OR "surrogate model" OR "machine learning")

This group of keywords was mainly used to supplement studies on traditional steel section optimization, metaheuristic optimization, and surrogate-assisted optimization, thereby providing a basis for comparing and positioning RL methods.

The keyword search yielded 4604 records from the two keyword groups described above. After the results were merged and 181 duplicate records were removed, 4423 records were retained for title and abstract screening. The intentionally broad search retrieved many records outside the review scope, commonly involving structural analysis and AI-based performance assessment rather than design and optimization. Title and abstract screening therefore excluded 4241 records. The remaining 182 records were then assessed through full-text screening, after which 137 were excluded because they lacked sufficient methodological relevance or did not address an optimization or design decision-making problem. To improve coverage, backward and forward citation tracing and manual supplementation were also conducted. Ultimately, 97 papers were selected for inclusion in the review.

Figure 2 summarizes the major source journals with more than two publications among the selected journal articles. Based on the current literature set, the reviewed papers were mainly published in journals related to structural engineering, computational mechanics, civil engineering informatics, and intelligent construction. Among them, Engineering Structures, Journal of Building Engineering, and Computer-Aided Civil and Infrastructure Engineering account for relatively large proportions. This distribution suggests that the reviewed literature remains centered on structural engineering applications, in which AI methods are commonly employed for damage identification [32], response prediction [3], candidate design generation [33], and structural optimization [7]. Recent research has further examined the use of AI in steel design, including inverse learning, uncertainty-aware assessment, and explainable decision-making [34].

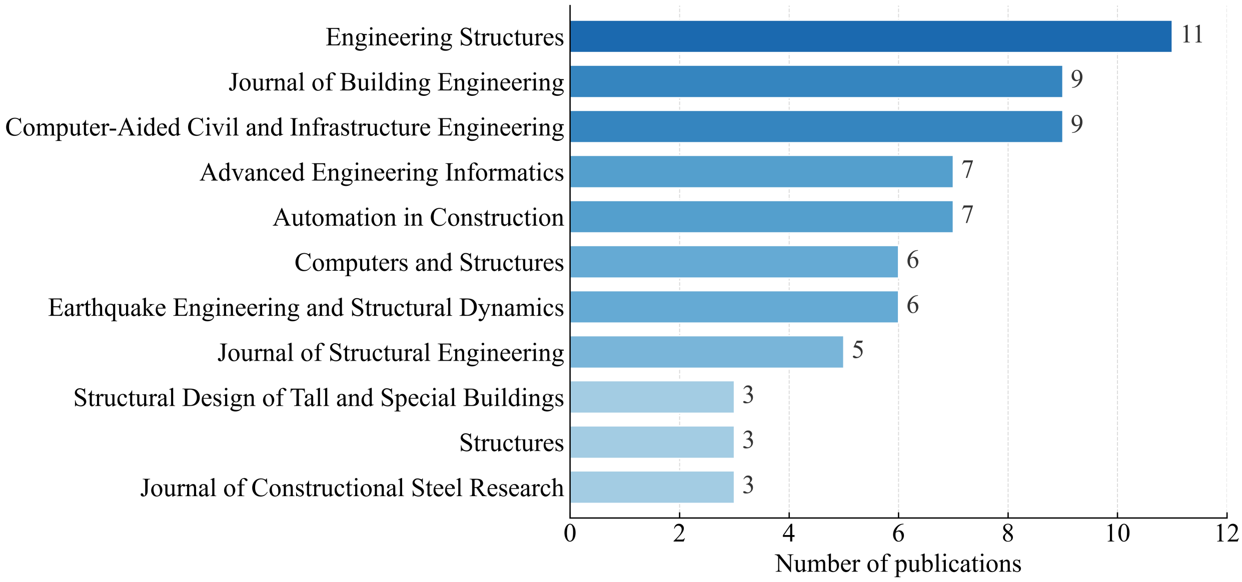

Figure 2. Journals with more than two reviewed publications.

3. Optimization problems for steel structures

3.1. Optimization task definition

In this review, optimization tasks for steel structures are generally described as constrained optimization problems. For a given structural model, the design-variable vector is selected to optimize an objective function while satisfying admissible design-space requirements and design constraints:

$$\begin{aligned} \min_{x} \quad & f(x), \\ \text{s.t.} \quad & x \in \Omega, \\ & g(x) \leq 0, \end{aligned} \tag{1}$$

where x denotes the design-variable vector, Ω is the admissible design space, $f(\cdot)$ is the objective function, and $g(\cdot)$ represents a set of constraint functions. The design space Ω covers section pools, grouping rules, topology or layout choices, geometric parameters, and other design restrictions, whereas $g(\cdot)$ collects code-compliance and performance-based requirements.

This formulation provides a unified basis for describing different optimization tasks for steel structures. In the following subsections, the main task types are distinguished by the definition of x, the admissible design space Ω, the objective function $f(\cdot)$, and the constraint functions $g(\cdot)$.

3.2. Task types and design variables

From the perspective of design variables, intelligent optimization of steel structures can be categorized into three main task types: section optimization, topology optimization, and the optimization of detailing and device parameters. These tasks differ in the definition of the design vector x, the admissible design space Ω, and the corresponding constraints, as summarized in Table 1.

Table 1. Taxonomy of intelligent design tasks and variables in steel structures.

Task type	Main design variables	Representative objects	Main concern
Section optimization	Section indices or size parameters	Steel frames	Code-compliant material reduction
Topology optimization	Component existence and connectivity	Steel-braced frames, trusses, and schematic layouts	Load-transfer rationality and stability
Optimization of detailing and device parameters	Shape, detailing, and device parameters	Spatial structures and seismic-control systems	Component feasibility and system performance

3.2.1. Section optimization

Section optimization is the primary focus of this review. For steel moment-resisting frames and steel-braced frames, the structural layout is typically determined during the schematic design stage, and subsequent optimization is therefore more commonly conducted at the component or component-group level. Under prescribed structural topology and loading conditions, the design vector x represents the selected sections or size parameters of sets of beams, columns, or braces. The objective function $f(\cdot)$ is commonly associated with steel consumption or overall construction cost, while the constraint functions $g(\cdot)$ correspond to the relevant strength, stability, stiffness, drift, and detailing requirements. In practice, section optimization is usually carried out as an iterative process involving scheme generation, structural analysis, code-compliance checking, and design revision, as illustrated in Figure 3.

Discrete section selection is the most common design-variable formulation in the optimization of steel structures. It involves selecting section labels for components from the predefined section pools. Since these section pools are discrete, and beams, columns, and braces usually correspond to different candidate section sets, section selection is inherently a combinatorial optimization problem [5,8,28,35].

In practical optimization, section selection is often modeled together with component grouping, rather than treating each component as a fully independent design unit. Components may be grouped according to story level, span direction, mechanical function, structural location, or engineering convention, and components within the same group are usually required to adopt identical or similar sections. Such grouping strategies can substantially reduce the dimensionality of the design variables, improve the regularity of section arrangements, and enhance constructability. They also reflect common engineering practice, in which sections are adjusted by typical stories, component types, or predefined component groups [6,7,29,36]. However, grouping also restricts the searchable design space and may reduce the potential for fine-grained component-level optimization. Therefore, component grouping is better understood as a strategy for variable organization and constraint embedding within section optimization, rather than as a task independent of section selection.

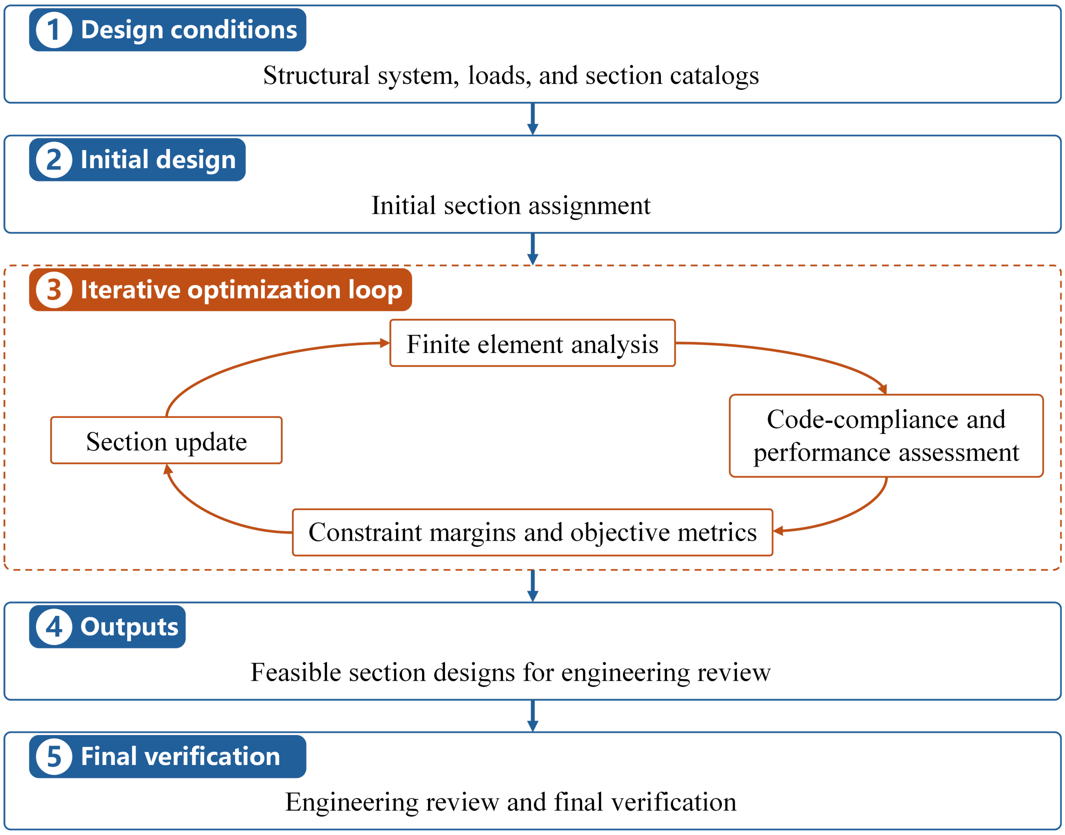

Figure 3. Workflow of steel section optimization.

In addition to discrete section selection, some studies incorporate continuous size parameters into the design variables, such as plate thickness, flange width, and section depth. These variables provide a more refined description of component geometry and connection parameters [9,37]. In practice, these continuous parameters are often further restricted by standard section series and fabrication rules, and are therefore mapped to discrete candidate sets or standardized values.

Recent data-driven studies have also explored the direct prediction of component dimensions, suggesting that such predictors can provide rapid initial size candidates that are later checked through structural analysis and code-compliance verification [38].

Section optimization can therefore be viewed as a constrained selection problem that links component-level section choices with system-level structural responses and code-compliance requirements.

3.2.2. Topology optimization

Compared with section selection, topology optimization operates at a higher design level. In this task, the design vector x mainly describes component existence, connectivity, bracing arrangements, column grids, or beam-system configurations, because these layout decisions directly affect structural load-transfer paths, stiffness distribution, and overall performance. The admissible space Ω is therefore governed by architectural layouts, stability requirements, and constructability requirements. Early studies on bracing optimization in steel frames incorporated brace type, brace location, and connection properties as combinatorial design variables, and used scatter search to obtain improved layouts satisfying displacement, strength, and limit-state constraints [4]. In more general structural topology design problems, truss and frame structures are often represented as graphs composed of nodes and edges, and the optimization process can be formulated as the sequential addition or removal of components, activation of nodes, or modification of connectivity within a predefined structural domain [39–42].

Schematic structural layout design is a related but broader task, because the initial structural layout often needs to be generated from architectural information rather than optimized within a fixed topology. Existing studies have generally followed two routes. The first route formulates this task as an image-to-image generation problem, where architectural plans are encoded as semantic images and structural layouts are generated as image-like outputs. This formulation has been widely used in schematic structural generation, including shear-wall, frame, and steel-braced frame layout design [43–45]. For steel-braced frames, Fu *et al.* adopted a two-stage generative adversarial network (GAN) [46] framework to generate column layouts first and bracing layouts subsequently. The generated schemes were then compared with engineer-designed layouts and further evaluated in terms of material consumption and finite element responses [33]. The second route formulates schematic structural design as a graph prediction problem. Under this formulation, architectural or structural plans are converted into graph representations, and the model predicts the categories of structural elements or the connectivity relationships among them [47–49]. Depending on whether components are represented as graph edges or nodes, the task can be further formulated as edge-level prediction or node-level prediction [50–52].

Beyond the geometric information contained in architectural drawings, structural layout generation also requires the incorporation of design conditions. Recent studies have introduced building height, seismic demand, structural type, and text prompts as conditional inputs to guide the generation process [53–56]. Meanwhile, because engineering datasets are often limited and structural design problems are strongly constrained, purely data-driven generation is generally insufficient. Existing studies have therefore improved generation feasibility by embedding design knowledge or physical rules into the modeling process [57–61]. Semi-supervised learning has also been introduced to alleviate the dependence on large labeled engineering datasets [62]. Other studies reduce the learning difficulty by decomposing the design process into staged subtasks, such as sequentially generating different structural components or combining layout generation with subsequent correction [33,63,64].

From a technical perspective, a key issue in applying AI to schematic structural design lies in the choice of structural representation. Early image-to-image methods commonly used semantic color maps to distinguish different architectural and structural components [33,43]. Subsequent studies adopted more explicit feature encoding, such as one-hot encoding, to reduce representation ambiguity and improve the consistency of layout generation [65]. Compared with image-based representations, graph-based representations are theoretically better suited to irregular structural plans because they can explicitly describe connectivity. However, existing studies also indicate that modern diffusion architectures can achieve strong performance in schematic structural generation tasks [54–56,66]. This suggests that diffusion models possess strong generative capability, whereas graph-based methods may require larger datasets and more appropriate structural representations to fully exploit the advantages of graph neural networks.

Compared with section selection, topology optimization involves a higher degree of design freedom and operates closer to the conceptual design stage. Although AI-based generative methods can rapidly produce diverse structural layouts, the generated schemes still require subsequent engineering verification of structural rationality and code compliance. Therefore, these methods should generally be regarded as tools for generating preliminary candidate schemes rather than directly applicable final designs.

3.2.3. Optimization of detailing and device parameters

Beyond section selection and layout design, intelligent structural design may also involve geometric form parameters, detailing rules, construction processes, and device-related parameters. In such tasks, the design vector x is more task-specific and may represent nodal coordinates, control points, component geometry, joint details, or seismic-control device parameters. The admissible space Ω is correspondingly governed by geometric feasibility, fabrication requirements, detailing provisions, and allowable parameter ranges. Such variables have been explored in studies that formulate structural design as a sequential decision problem for trusses, shells, and other spatial structures [67–69]. In steel and composite structural systems, related studies have further extended intelligent optimization to component-level and detailing-related variables, such as composite beam parameters, joint configurations, and connector layouts [70–72]. For steel components with more specialized geometric features, inverse-learning formulations have also been explored to directly infer section and material parameters from target resistance and geometric descriptors, as demonstrated in recent studies on perforated steel beams [73]. For seismic or vibration-control applications, brace systems, dampers, and other energy-dissipation devices may introduce additional design variables associated with stiffness, strength, damping, and reliability requirements [74,75]. Although these tasks further expand the design space of intelligent structural optimization, their variables and constraints are more heterogeneous and task-specific than those in section optimization.

4. Traditional optimization methods for steel structures

For steel structures, section optimization is one of the most extensively studied tasks in structural optimization research. For steel moment-resisting frames and steel-braced frames, once the structural system, loading conditions, and section pools are specified, the optimization task is typically to assign appropriate sections to different components or component groups, so as to reduce material consumption or overall cost while satisfying strength, stability, stiffness, and detailing requirements.

Traditional studies have formulated this task as a discrete search problem over section assignments. On this basis, optimization algorithms are often coupled with component grouping strategies, rule-based correction procedures, and code-compliance checks to improve engineering applicability. Surrogate models are introduced when repeated finite element analyses or performance evaluations become computationally expensive. In this sense, traditional methods serve as baseline search strategies for steel section optimization and provide practical mechanisms for constraint modeling and computational acceleration.

4.1. Traditional optimization algorithms

Metaheuristic algorithms have long served as major solution tools for steel section optimization. These methods typically treat structural design as a black-box optimization problem, in which candidate designs are generated, evaluated, and updated to search for improved section combinations within a prescribed design space.

Early studies applied SA to the discrete optimization of three-dimensional steel frames. Balling formulated steel-frame design as a minimum-weight optimization problem under seismic and gravity loads, considering constraints on interstory drift, combined stresses, and section geometry. The method reduced the risk of premature convergence to local optima by probabilistically accepting non-improving solutions, and adopted approximate analysis to accelerate structural response evaluation [35]. SA is well-suited to discrete combinatorial problems and can be conveniently combined with approximate analysis to reduce computational cost. In such applications, however, the final optimization performance may still be affected by the initial design, algorithmic parameter settings, and the accuracy of the approximation model, which are common sensitivity issues in traditional optimization workflows.

GA [14] subsequently became widely used in the optimization of steel structures. Adeli *et al.* explored the application of parallel GA to large space steel structures, improving the efficiency of large-scale structural optimization through parallel evaluation of population individuals [37]. Liu *et al.* applied a GA to the minimum-weight seismic design of steel frames and incorporated design complexity into the objective function to control the number of different sections and improve constructability [1]. These studies broadened the optimization of steel structures from pure weight minimization to more engineering-oriented designs that consider section regularity, detailing simplicity, and constructability.

Traditional section-combination search has also been extended to higher-level discrete decisions, such as brace placement, brace type, and connection configuration. Hagishita *et al.* used scatter search to optimize brace types, brace locations, and connection properties in steel frames, while accounting for the influence of semi-rigid joint behavior on structural performance [4]. Compared with pure section optimization, such problems directly affect load-transfer paths, global stiffness, and structural stability, and therefore rely more heavily on prior design rules, candidate screening, and subsequent structural verification.

The introduction of swarm intelligence algorithms further expanded the methodological range of optimization studies on steel structures. Hasançebi *et al.* employed a bat-inspired algorithm for code-compliant minimum-weight design of steel frames, and compared its performance with other heuristic methods using examples containing thousands of components [76]. Li *et al.* proposed a section optimization method for complex long-span spatial steel structures based on internal force states and the grey wolf optimizer. In their method, component stress ratios, nodal displacements, and adaptive section update coefficients were incorporated into the section updating process to reduce steel cost while satisfying strength, stability, and global stiffness constraints [9]. Shan *et al.* integrated the interpretation of architectural computer-aided design (CAD) plans, parametric modeling, and an improved PSO algorithm to establish an automated workflow from building plans to steel frame models and section optimization [77]. These studies show that swarm intelligence algorithms have been widely explored as search strategies for constrained section optimization, while recent workflow-oriented studies have incorporated them into broader pipelines linking model generation, structural analysis, and design verification.

Beyond metaheuristic search, parallel computing and neural dynamics models have been explored to improve the computational efficiency of large-scale optimization of steel structures. Park *et al.* proposed a neural dynamics model for plastic design optimization of steel frames, using Lyapunov stability theory, external penalty functions, and variable normalization to handle constrained optimization problems [78].

Distributed neural dynamics and parallel neural dynamics models were later applied to large spatial steel structures and super-tall steel structures, with design-code-based checks on stresses, displacements, and fabrication constraints [79,80]. These studies also reflected an early recognition that, beyond the search strategy itself, the efficiency of structural analysis, constraint checking, and candidate evaluation can substantially influence the computational performance of large-scale optimization of steel structures.

Beyond standalone search algorithms, some studies have placed greater emphasis on aligning section optimization workflows with engineering design logic. Wan *et al.* proposed a two-stage discrete sizing optimization method for steel frames, in which global search based on the DIviding RECTangles (DIRECT) algorithm and local response-surface approximation are combined to improve search efficiency, followed by local correction of components that violate strength or stability requirements after global performance criteria are satisfied [5]. Jin *et al.* incorporated component grouping and standardized section indices into a response-surface-based optimization workflow for high-rise steel–concrete composite buildings, improving the efficiency and practical applicability of section optimization for complex high-rise structures [36]. These studies indicate that, although improving the search capability of general optimization algorithms can enhance performance, incorporating engineering assumptions and design practices into the optimization workflow can also effectively improve efficiency, feasibility, and practical applicability.

Traditional optimization algorithms can be integrated with finite element analysis and code-compliance checking, with engineering constraints addressed through penalty, screening, and correction mechanisms. They therefore remain an important technical foundation and comparison baseline for section optimization.

4.2. *Surrogate-assisted optimization*

The computational cost of structural optimization mainly arises from repeated analysis and verification of candidate designs. The introduction of surrogate models does not change the basic formulation of the optimization problem; rather, it aims to replace or screen the most expensive evaluation steps. According to their roles, surrogate models can be broadly divided into two categories. One category predicts structural responses or performance indicators, such as displacement, internal force, interstory drift ratio, damage index, failure probability, and life-cycle cost. The other evaluates the feasibility or constraint status of candidate designs, thereby reducing finite element analyses for clearly ineffective or infeasible solutions. Both categories aim to improve optimization efficiency while maintaining sufficient reliability. Related reviews have similarly positioned surrogate models, physics-informed learning, inverse methods, and explainable AI as tools for efficient structural analysis and decision support, rather than replacements for physics-based verification [81].

Response-prediction surrogates are the most common form. Their primary role is to approximate structural responses or damage indicators at a lower computational cost. In the reviewed studies, response-prediction surrogates are mainly based on artificial neural networks (ANNs) and their variants, as well as probabilistic models such as Bayesian linear regression (BLR), Gaussian process regression (GPR), and Kriging. Gomes *et al.* combined PSO with ANN surrogates for global risk optimization of a steel transmission tower with 44 design variables. Multilayer perceptron (MLP), radial basis function (RBF), and combined MLP–RBF surrogates were used to approximate the risk-optimization objective function. Compared with direct global optimization, the surrogate-assisted strategies reduced computational time by 20%–69% and avoided 22%–48% of mechanical model calls, although the results were sensitive to the surrogate type and training-set size [11]. Gholizadeh *et al.* compared RBF, generalized regression (GR), feed-forward back-propagation (FFBP), and cascade-forward back-propagation (CFBP) neural networks for predicting maximum interstory drift ratios and global damage indices of 6- and 12-story steel moment-resisting frames. The RBF model showed the best generalization performance and was embedded into the optimization loop to avoid repeated nonlinear time-history analyses under 10 earthquake records for each candidate design [2]. In subsequent work, an ensemble of parallel neural networks was used to predict interstory drift, floor acceleration, and global damage indices at multiple performance levels, thereby avoiding the need to perform 36 nonlinear time-history analyses for each candidate design during optimization [3]. Motamedi *et al.* used BLR to construct probabilistic drift-demand models from incremental dynamic analysis data and coupled them with a learning-based variant of charged system search for seismic loss optimization of 6-, 12-, and 18-story steel frames. The regression model reduced the cost of repeated seismic loss evaluation, while the learning-based optimizer accelerated convergence relative to the standard charged system search [82]. These studies show that response-prediction surrogates can substantially reduce nonlinear analysis or loss-evaluation costs, but their reliability still depends on training-data coverage and surrogate selection, while validation remains limited to a small set of structural examples.

For optimization problems governed by reliability constraints, multi-objective trade-offs, and nonlinear dynamic responses, a response surrogate trained only once may be insufficient to ensure reliable optimization. Therefore, many studies have adopted sequential sampling strategies that identify and add informative samples during the optimization process. Do *et al.* used GPR to approximate nonlinear time-history responses in multi-objective seismic reliability optimization of steel moment-resisting frames, and concentrated new samples near the Pareto front and feasible-domain boundaries, thereby handling discrete variables, multiple objectives, and probabilistic constraints within a limited analysis budget [13]. Xiao *et al.* applied GPR and constrained expected improvement to parameter optimization of mega-sub controlled structures and mega-frame structures, reducing time-history analysis calls and improving seismic performance by prioritizing samples with high improvement potential or constraint sensitivity [83]. Peng *et al.* introduced Kriging surrogates and expected improvement into real-time hybrid simulation-driven reliability optimization; when numerical models could not fully capture complex nonlinear behavior, experimental feedback was used to update the surrogate model and improve design reliability [84]. These studies show that surrogate models reduce the cost of individual response evaluations and guide sample enrichment through uncertainty measures or improvement criteria, thereby allocating high-cost analyses to feasible-domain boundaries and other critical regions.

In section optimization, surrogate models are commonly embedded into metaheuristic or multi-objective evolutionary algorithms to accelerate candidate evaluation, while prediction errors are progressively corrected through high-fidelity analysis and sample enrichment. Shan *et al.* proposed an online surrogate-driven section optimization workflow for steel frames, combining support vector regression, deep neural networks, and light gradient boosting machine (LightGBM) with GA and PSO; online training and the backfilling of potentially high-quality individuals were used to accelerate convergence [6]. Wang *et al.* proposed a multi-objective optimization framework for super-tall buildings that combined a generalized regression neural network (GRNN) surrogate with the non-dominated sorting genetic algorithm II (NSGA-II), where GA was used to optimize the smoothing factor of the GRNN. Pareto solutions obtained from the surrogate model were selectively verified through simulations or experiments, and inaccurate samples were added back to the training set for iterative surrogate updating [85]. Huang *et al.* introduced a neural network surrogate for frequency prediction in the optimization of tall buildings under wind and earthquake hazards, and combined it with an improved NSGA-II and optimality criteria to improve the efficiency of the multi-objective search [86]. Truong *et al.* incorporated a multi-objective comparison technique and an extreme gradient boosting (XGBoost) surrogate into NSGA-II for bi-objective optimization of nonlinear steel trusses, using an adaptive safety parameter to reduce misjudgments caused by surrogate error [17].

In addition to regression-based response surrogates, feasibility classification has also become an important direction in recent structural optimization studies. For strongly constrained optimization problems, many candidate designs are evidently infeasible at an early stage, and full finite element analyses for all such designs can cause substantial computational waste. By predicting feasibility rather than detailed responses, classifiers simplify the prediction task and can serve as screening tools, while finite element analysis is retained for quantitative evaluation and verification. Cao *et al.* proposed a feasible-domain boundary identification method based on virtual sampling and support vector machines, in which constraint checking was transformed into a feasible-or-infeasible classification problem. During optimization, the classifier was used to rapidly screen candidate designs, and high-fidelity structural analysis was performed only for critical cases [87]. Cao *et al.* further developed a vertex-based graph neural network for feasibility classification in truss section optimization, where components were represented as graph vertices and topological information was explicitly incorporated, thereby reducing the number of structural analyses while maintaining design quality [18]. Prayogo *et al.* used LightGBM-based surrogate models in structural optimization to screen infeasible or low-potential candidates before high-fidelity analysis, with the surrogate updated using newly evaluated samples during the search process [88]. These studies extend the role of surrogate models from response approximation to feasible-domain identification and candidate screening, providing an effective means of reducing redundant analyses in strongly constrained structural optimization.

Beyond their roles in response prediction and feasibility screening, surrogate models have also evolved in how structural data are represented. Conventional models often use design-variable vectors as inputs [89], but structural responses are also governed by response histories, nodal connectivity, component topology, and spatial relationships across structural scales. This motivates the use of sequence-based and graph-based representations. Accordingly, recent studies have introduced sequence-based seismic response prediction and neural-network-assisted optimization to improve response evaluation and support efficient design search across related structural instances [90,91]. Kuo *et al.* represented steel moment frames as graph structures composed of beam-column nodes and components, and combined graph neural networks with long short-term memory (LSTM) to predict nonlinear seismic response histories of frames with different numbers of stories and spans [92]. Zhang *et al.* proposed a graph-deep-learning-based differentiable optimization framework for structural section design, in which the graph surrogate model predicts interstory drift ratios and story lateral stiffness from structural layouts and section variables, enabling gradient-based updates within the optimization process [93]. Fei *et al*. integrated graph neural networks with evolutionary algorithms for rapid optimization evaluation of shear-wall structures at the schematic design stage, and used a distance-based update strategy to correct surrogate errors [94]. These studies improve the accuracy of individual response predictions while incorporating structural topology and component relationships into surrogate modeling, thereby supporting rapid evaluation across different structural scales, topologies, and layouts. However, surrogate models should be viewed with caution. Although they provide an effective route for improving efficiency during optimization, their development still requires careful model design, data collection, and training, and their influence on the reliability of optimization results remains to be systematically verified.

Surrogate models can reduce the demand for high-cost analyses and improve the computational tractability of complex structural optimization tasks, including performance-based seismic, reliability-based, and life-cycle cost optimization. However, their reliability depends on the coverage of training samples and the control of prediction errors near critical constraint boundaries, while model development still requires substantial high-fidelity analysis data.

4.3. Constraint modeling and multi-objective optimization

Constraint modeling and multi-objective treatment are central to the optimization of steel structures. A design with a favorable objective value cannot be regarded as engineering-feasible unless it satisfies code requirements and constraints related to structural feasibility and constructability. At the same time, optimization tasks for steel structures are rarely governed by a single objective. Material consumption, construction cost, structural performance, and sustainability-related indicators often need to be balanced within the same optimization process.

Code-based constraints are the most direct source of design requirements. They include strength and stability checks, interstory drift limits, panel-zone checks, width-to-thickness and slenderness limits, seismic detailing provisions, and load-combination rules under different limit states. Traditional optimization studies on steel frames have incorporated design code provisions into section selection by checking stress limits, displacement limits, buckling resistance, and detailing requirements [1,35,76].

Structural feasibility and constructability further constrain the design space, especially when layout and connection choices are treated as discrete variables. Hagishita and Ohsaki applied a modified scatter search method to steel frames with semi-rigid joints, in which brace types, brace locations, and semi-rigid joint types were optimized as discrete design variables and evaluated using refined plastic hinge analysis [4]. Constructability requirements are usually reflected through component grouping, controlled section diversity, and regular section transitions. For example, design-complexity terms or component-grouping strategies have been used to promote regular and practical section arrangements [1,5].

From an algorithmic perspective, constraint modeling can be broadly classified into penalty-based methods, repair methods, and filtering or prior-screening methods. Penalty-based methods convert constraint violations into penalty terms in the objective or fitness function, and are widely used in metaheuristic optimization [6,76]. Their effectiveness depends strongly on penalty scaling. Weak penalties may allow infeasible solutions to persist, whereas overly strong penalties may drive the search toward conservative feasible regions and limit the discovery of better feasible designs. Repair methods correct infeasible designs through rule-based adjustment or local search, such as enlarging components that violate strength or stability requirements after global performance criteria are satisfied [5]. Filtering and prior-screening methods exclude invalid or low-potential candidates before expensive structural analyses. Feasibility classifiers and surrogate screening models have been used for this purpose by identifying feasible-domain boundaries or screening candidates before high-fidelity evaluation [87,88]. These strategies can improve feasibility and reduce the search space, but overly restrictive rules or inaccurate screening models may also exclude potentially high-quality solutions.

In the optimization of steel structures, objective functions and constraints are often closely coupled. In many section optimization formulations, economy-related indicators, such as material cost and life-cycle cost, are used as primary objectives, whereas strength, stability, stiffness, and detailing requirements are imposed as constraints. Early optimization studies on steel frames commonly focused on minimum-weight design subject to strength, displacement, and detailing constraints [35,76]. With advances in performance-based, reliability-based, and resilience-oriented design, optimization targets have expanded beyond material reduction to include seismic loss, repairability, and expected life-cycle cost [2,3,11]. These indicators may be treated either as additional objectives or as constraints, depending on the formulation.

Multi-objective treatment is commonly implemented in three ways. The first is scalarization, in which multiple objectives or indicators are combined into a weighted scalar objective. This approach is simple and suitable for constructing integrated objective or fitness functions, but the results are sensitive to weight selection, normalization, and penalty scaling [94]. The second is constraint-based formulation, where one primary objective, such as steel consumption, cost, or carbon emissions, is minimized while drift limits, strength checks, stability requirements, detailing rules, and code provisions are imposed as constraints [5,6]. The third is Pareto approximation, which provides a set of non-dominated solutions for subsequent engineering selection. This approach is useful for problems with conflicting objectives and uncertain design preferences, although it is computationally more demanding and still requires project-specific judgment [10,13].

For traditional and surrogate-assisted optimization, constraint modeling and multi-objective formulation jointly provide the basis for constructing integrated evaluation criteria. Constraint satisfaction, material efficiency, and performance improvement must be balanced carefully; otherwise, optimization may favor low-weight but infeasible designs or overly conservative feasible solutions.

4.4. Limitations of traditional methods

Although traditional optimization algorithms and surrogate-assisted methods have provided a mature foundation for steel section optimization, they are usually organized around individual design cases. Metaheuristic algorithms search and update candidate section combinations within a given structural model, while surrogate models reduce the cost of repeated response prediction or feasibility evaluation in a prescribed design domain. These mechanisms can improve the efficiency of single-case optimization, but they do not explicitly learn reusable decision policies for adjusting sections across related structural instances. This limitation becomes important when steel moment-resisting frames and steel-braced frames are considered as structural families rather than isolated examples. Related structures may differ in scale, layout, loading conditions, and section pools, yet they often share similar component roles, load-transfer mechanisms, and code-compliance logic. This motivates the development of more generalizable optimization methods that can reuse design experience across related structural instances. RL provides a suitable formulation for this goal by learning reusable section adjustment policies through repeated interactions with structural environments, as illustrated in Figure 4.

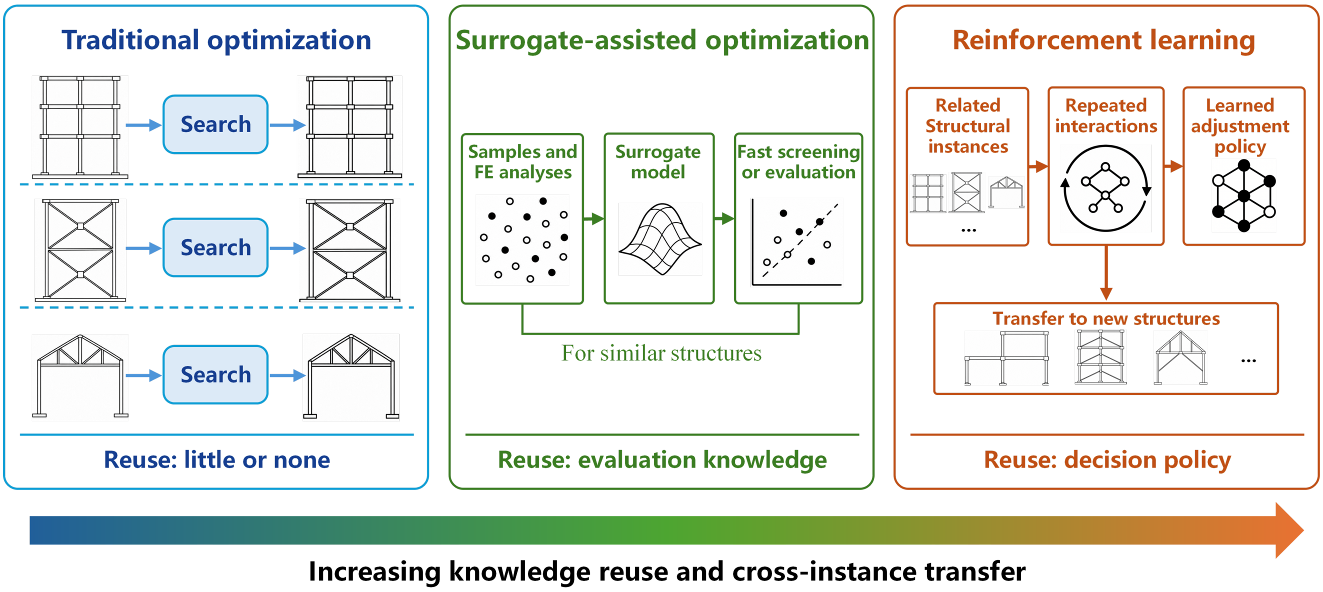

Figure 4. From case-specific search to reusable policy learning.

5. RL-based optimization for steel frames

This section discusses RL-based optimization for steel frames from the perspective of structural optimization modeling. The discussion focuses on task formulation, sequential decision modeling, reward function design, RL algorithms, and evaluation settings and baselines. Emphasis is placed on constraint-aware learning and policy reuse across related structural instances.

5.1. Motivation for RL-based structural optimization

The main purpose of introducing RL into structural optimization is to obtain an optimization policy that can guide design decisions through policy or value learning. Under this formulation, the agent learns how to modify structural sections through interactions with the structural environment, rather than relying solely on repeated task-specific search. Once the policy has been trained, the deployment stage can use the learned strategy to guide design modifications with fewer trial evaluations.

However, the efficiency of RL-based optimization should be evaluated with caution. If an RL method is trained and applied only to a single structural instance, both training time and inference time should be included in the total computational cost of solving that optimization task [95]. Since RL is often sample inefficient, training may require a large number of finite element analyses and code-compliance evaluations. In this case, RL may be less efficient than conventional metaheuristic or surrogate-assisted optimization.

The core value of RL-based structural optimization, therefore, lies in the reuse of policies across related structural instances. The training stage aims to acquire a reusable optimization policy, while the deployment stage uses the trained policy to solve new structural instances with fewer structural evaluations. Under this setting, training and inference costs can be discussed separately. The training cost becomes acceptable only when the learned policy generalizes across a sufficiently broad structural domain.

Current studies can be organized into three routes for achieving cross-structure policy reuse. The first route is pretraining followed by few-shot fine-tuning. A policy is first trained on representative structural instances and then adapted to a new task with limited additional interactions [95,96]. The main challenge is to keep the fine-tuning cost sufficiently low so that the combined cost of fine-tuning and deployment remains competitive with heuristic optimization. The second route is direct transfer based on generalizable structural representations, in which the policy is trained on part of the structural domain and then directly applied to unseen structures. Graph-based representations have been used to handle variations in structural scale and connectivity [8,39,97]. The third route is to formulate structural optimization as a multi-task RL problem. Different structures are regarded as samples from a task distribution, and the policy is conditioned on structural context, such as structural scale, layout, loading condition, component organization, and section pools [7,98]. This formulation corresponds to a contextual Markov decision process (cMDP) view, in which large-scale pretraining is expected to produce a more general optimization policy.

For the second and third routes, the trained agent is expected to optimize new structures without task-specific fine-tuning. In this setting, the deployment-stage efficiency advantage over traditional search is more direct. The training cost can then be regarded as a preparatory investment for building a general optimization capability, rather than as the optimization cost of a specific structural instance. However, this interpretation is justified only when the policy generalizes reliably within a clearly defined structural domain. Therefore, the training and testing domains should be explicitly defined in terms of structural instances, design variables, and candidate design spaces.

Beyond deployment efficiency, RL may also improve optimization quality by changing how the search process is organized. Through action-space design, a structural optimization problem can be reformulated from direct search over complete design schemes into sequential decisions over design adjustments. This decomposition can reduce the difficulty of exploring a high-dimensional combinatorial design space. In addition, policy learning across related structures exposes the agent to a distribution of optimization tasks. Strategies that only exploit local features of a specific structure are less likely to remain effective across the training domain, which may help the learned policy avoid case-specific local optima. These considerations indicate that the motivation for RL-based structural optimization lies in reducing online search through reusable policies and improving search guidance through sequential decision modeling.

5.2. RL formulation for structural-family optimization

Following the MDP formulation introduced in Section 1, RL-based optimization for a fixed structural instance can be described as an agent-environment interaction process. At step t, the current structural design is represented by the state s_t, and a section modification is selected as the action a_t. Finite element analysis and code-compliance checking then determine the transition to s_{t+1} and provide the reward

$r_t = R(s_t, a_t, s_{t+1})$. Through repeated interactions, the agent learns a policy for improving the current design according to analysis-based feedback.

For steel frame section optimization, the state usually includes current section assignments, component or component-group information, structural responses, and code-check margins. The action space defines allowable section modifications, such as increasing, decreasing, retaining, or selecting section indices from candidate pools. These elements have been adopted in studies such as FrameRL, FrameGym, MAFO-3D and FrameMARL for steel-braced frames [7,28,29,98]. Figure 5 illustrates this basic MDP formulation.

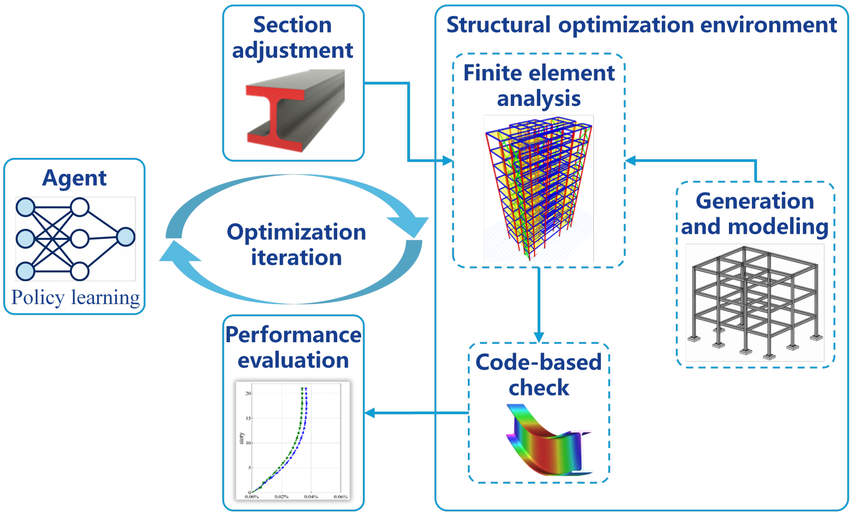

Figure 5. RL formulation for section optimization of steel frames.

When the objective shifts from optimizing a single structure to learning reusable policies across a structural family, the MDP formulation can be extended to a contextual Markov decision process (cMDP). In this setting, each structural instance is associated with a context variable c, which describes task-level information such as structural scale, layout, loading condition, section pools, and design constraints. The structural-family optimization problem can then be formulated as:

$$M_c = (S_c, A_c, P_c, R_c, \gamma), \quad c \in C, \tag{2}$$

where C denotes the context space of the structural family, S_c and A_c denote the state and action spaces under context c, $P_c(s_{t+1} \mid s_t, a_t)$ denotes the context-dependent transition function, and $R_c(s_t, a_t, s_{t+1})$ denotes the corresponding reward function. The policy is written as $\pi(a|s,c)$, meaning that action selection depends on both the current design state s and the structural context c.

Under the cMDP formulation, generalization depends on how the context space is defined and sampled. Training structures define the observed context domain, while testing structures evaluate whether the learned policy can be transferred to unseen contexts. The Maximum Considerable Structure (MCS) provides a representative way to define such a structural-family domain for steel frame section optimization by specifying the range of structural scales, configurations, component organization, and design-variable spaces [7].

This formulation makes state and action representations central to structural-family optimization. For a single MDP, the state and action spaces can be tailored to one fixed structure. For a cMDP, they must remain compatible across different structures. Tensor representations are suitable for regular frame systems whose stories, spans, component types, and verification results can be mapped to structured arrays [7,98]. Graph representations are more suitable for variable-scale or irregular systems because they explicitly describe nodes, components, and connectivity [8,39]. In both cases, section indices, structural responses, and constraint margins should be organized in a scale-consistent form, so that the policy can learn comparable decision rules across structural instances.

The formulation of RL-based structural optimization can therefore be understood at two levels. The basic MDP describes how an agent improves one structural design through sequential interaction with the analysis and design-check environment. The cMDP extends this formulation to a structural family by introducing context variables and shared policy learning, thereby providing the modeling basis for policy reuse and cross-structure generalization.

5.3. Reward function design

Reward design in RL-based structural optimization should be consistent with both the optimization objective and the action paradigm. For a structural optimization task, a direct formulation is to provide a terminal reward after a complete design has been generated and verified. The reward can then be defined according to final material consumption, structural performance, and code-compliance status. This formulation is closely aligned with the original optimization objective, because section optimization is ultimately evaluated by the verified final design. However, terminal rewards are often sparse, especially when the agent needs multiple adjustment steps before reaching a feasible and economical scheme.

A common way to improve trainability is to introduce intermediate rewards based on constraint margins. Compared with a binary feasible-or-infeasible signal, continuous constraint margins quantify the distance between the current design and the code-compliance boundary. This allows the agent to distinguish severe violations from near-feasible states and to receive informative feedback before a fully feasible design is obtained. For section optimization, such margins may be derived from stress ratios, stability checks, drift limits, and other code-based verification results. The reward can therefore guide the agent toward feasibility first and further encourage material reduction after feasibility is approached.

A potential-difference formulation provides a more structured form of constraint-margin-based feedback. Instead of using constraint margins directly as step rewards, a potential function can be constructed from performance indicators derived from these margins, and the reward can be defined by the potential difference between two consecutive states. MAFO-3D provides a representative example for steel frame section optimization [7]:

$$r_t = \Phi(s_{t+1}) - \Phi(s_t), \tag{3}$$

where $\Phi(\cdot)$ denotes a state potential associated with structural feasibility and performance. In this method, code-check results are first converted into safety indicators and then mapped into component performance values through a nonlinear function. These values are further organized into a structural performance vector in a magnetic-field space, and the reward is computed from the potential difference between states. In this sense, the magnetic-field-based potential function can be regarded as an enhanced constraint-margin reward, because it transforms component-level verification results into a denser learning signal.

For episodic structural optimization, reward shaping should remain aligned with the final design objective. MAFO-3D adds a terminal correction term so that the cumulative return reflects the highest structural potential reached within an episode rather than only the potential at the final step [7]. This treatment preserves dense intermediate feedback while aligning the return with the best design found along the trajectory.

Reward normalization is necessary when policies are trained across different structural instances. Structural weight, internal forces, drift responses, and the number of valid components may vary substantially with structural scale, loading condition, and layout. Directly using raw quantities may cause large structures to dominate the reward magnitude and weaken learning signals from smaller structures. Therefore, reward terms should be normalized by reference weight, code limits, component capacities, or the number of valid components. This is particularly important for structural-family training, where the same policy is expected to learn from structures with different numbers of stories, spans, and components.

Reward design should also reflect the action paradigm. For stepwise section adjustment, intermediate rewards are needed to indicate whether each action improves feasibility or design quality. Constraint-margin rewards and potential-difference rewards are suitable for this setting. For direct section assignment from candidate pools, one action may already produce a complete or near-complete design, so the reward can be based more directly on the evaluated design performance. Therefore, the action paradigm determines whether the reward should emphasize stepwise guidance or design-level evaluation.

MAFO-3D and MAGIC-Frame further demonstrate how global and local rewards can be coordinated under different decision formulations [7,99]. In MAFO-3D, local potentials are derived from the performance of individual substructures and normalized so that their sum equals the global potential. Local rewards are defined by changes in these potentials with a corresponding terminal correction, so that their cumulative sum across agents matches the global return. MAGIC-Frame adapts this principle to direct section assignment in a single step. It constructs global and local rewards from positive and negative potentials based on component and structural verification indices. The local positive potentials are normalized to sum to the global positive potential, while the global negative potential is defined as the sum of the local negative potentials. Consequently, the local rewards across agents sum to the global reward at each design attempt. In reward ablations with all other multi-agent group relative policy optimization (MAGRPO) settings held constant, using only the global reward accelerated the early learning of safe designs but did not support continued improvement in structural verification indices. Using only local rewards resulted in lower reward values and poorer training stability. The complete reward formulation supported a stable learning process in which the policy first learned to satisfy component checks and then progressively improved structural performance.

5.4. RL algorithms used in structural optimization

The choice of RL algorithms in structural optimization should be considered jointly with the task scale, structural representation, action space, reward design, and decomposition strategy. Steel frame section optimization provides a representative example for comparing these algorithmic choices. When section modification is formulated as a limited set of enumerable actions, such as increasing, decreasing, or retaining section sizes, value-based methods can be applied directly. Hayashi *et al.* used a deep Q-network (DQN) to assist metaheuristic search for story-wise section adjustment and subsequently adopted Q-learning with graph embedding to guide member-wise one-step section updates [8,100]. When section decisions are represented by a high-dimensional vector of policy outputs, policy-gradient actor–critic methods such as proximal policy optimization (PPO) can be adopted. In FrameRL, PPO outputs continuous action values that are mapped to standard beam and column section pools [98]. For larger steel structural systems with naturally distributed design units, section selection can be further formulated as a multi-agent problem, in which decisions are assigned to agents associated with stories, substructures, floors, or component groups. These examples indicate that Q-learning and DQN-type methods are directly aligned with enumerable discrete section adjustments, PPO-type methods can accommodate high-dimensional section-selection outputs, and multi-agent formulations are suitable for large-scale tasks with mechanically meaningful decomposition.

A more direct algorithm-level comparison is provided by FrameGym, which benchmarks seven RL algorithms using four steel section-sizing environments implemented within a common tensorized Generator–Solver–Evaluator framework [28]. These environments represent progressively more complex tasks: Comp sizes a single steel member; Multi-Comp simultaneously sizes multiple members (five in the reported experiments); (MRF-2D) optimizes a planar steel moment-resisting frame; and (CBF-2D) adds braces to form a planar concentrically braced frame with more components and more complex force-transmission paths. Not all seven algorithms were evaluated in every environment. In the FrameGym implementation, the authors evaluated dueling DQN and advantage actor–critic (A2C) only in Comp because applying these algorithms to vector-valued actions over the 155-section pool would lead to exponentially many joint section combinations and require substantially larger models. In Multi-Comp, PPO was less effective than the other four tested algorithms when the action directly selected a section index (a_{des}). By contrast, relative increments or decrements of the current section index (a_{opt}) improved all five algorithms by limiting the change in sectional properties at each step and thereby reducing instability during early training. In MRF-2D, deep deterministic policy gradient (DDPG) and twin delayed deep deterministic policy gradient (TD3), both of which use deterministic policies, exhibited insufficient exploration under component coupling and environmental noise, including section-pool noise arising because ordering sections by one property does not make their other mechanical properties vary monotonically. By contrast, the stochastic-policy methods trust region policy optimization (TRPO), PPO, and soft actor–critic (SAC) achieved better performance. However, a_{opt} no longer offered an unambiguous advantage in this environment: it produced higher rewards but less stable pass ratios than direct section selection. In the more difficult CBF-2D environment, where braces enlarge the action space and complicate force transmission, PPO was the only algorithm trained effectively under the tested settings. With a_{des}, SAC improved as the episode length increased but still failed to attain a pass ratio of 1, indicating that SAC did not consistently produce fully compliant designs. With a_{opt}, all algorithms converged to low reward levels. These results demonstrate that algorithm performance is highly sensitive to the action formulation, section-pool ordering, environmental noise, and structural coupling. Among the tested configurations, PPO showed the strongest robustness as task complexity increased to CBF-2D, although it was not uniformly superior in the simpler environments.

Beyond steel frame section optimization, PPO has also been applied to architectural space layout and truss topology and geometry optimization, where structured layout actions, topology modifications, and coordinate shifts are involved [101,102]. For tasks formulated directly with continuous geometric variables or other design parameters, deterministic actor–critic methods such as DDPG and multi-agent DDPG (MADDPG) have been adopted for latticed-shell geometry optimization and multi-objective truss optimization [10,103]. Table 2 compares representative RL formulations in structural optimization, including the structural task, state definition, action space, reward design, and adopted algorithm. The studies summarized in the table cover Q-learning, DQN, hierarchical deep reinforcement learning (HDRL), PPO, multi-agent PPO (MAPPO), MAGRPO, DDPG, and multi-agent DDPG (MADDPG).

Multi-agent RL is commonly introduced when an optimization task can be decomposed into multiple interacting design units. One typical case is large-scale structural optimization. For three-dimensional steel frames, section decisions involve many beams, columns, and braces, leading to high-dimensional states, large action spaces, and difficult credit assignment in a single-agent formulation. Another case is reinforcement arrangement, where spatially distributed rebars can be represented as interacting agents for path planning and clash resolution [104]. In these tasks, agents can be assigned to different stories, substructures, component groups, or rebars, thereby reducing the decision burden of each agent.

Existing steel frame studies mainly use structural decomposition to construct such multi-agent formulations. MAFO-3D divides a three-dimensional steel moment-resisting frame into story-level substructures and assigns section decisions to agents at the component-group level. Global states, local observations, global and local rewards, and a two-level critic are used to coordinate local decisions with the structural objective [7]. FrameMARL decomposes planar steel-braced frames into floor-level agents, with each agent selecting beam, column, and brace sections according to local structural information and finite element feedback [29]. These studies show that multi-agent RL can reduce the dimensionality of large structural optimization tasks and reflect engineering grouping rules. More recently, MAGIC-Frame extended this route to irregular three-dimensional steel moment-resisting frames by combining MAGRPO with structural- and component-level code checks [99].

The effectiveness of multi-agent RL depends on whether the decomposition is mechanically reasonable. Decomposition by stories, substructures, or component groups can simplify decision-making, but structural responses remain globally coupled. Interstory drift, global stability, structural period, internal force redistribution, and total steel consumption are jointly affected by multiple local actions. Therefore, agents responsible for local components or substructures must still receive sufficient global information to align their decisions with the structural objective.

Table 2. Comparison of RL formulations in representative studies.

Study	Structure/Task	State	Action space	Reward design	Algorithm
FrameRL [98]	Planar steel frames; section optimization	Component tensor: sections, loads, checks	Beam/column section-index selection	Code compliance and material efficiency	PPO
FrameMARL [29]	Steel-braced frames; section optimization	Global state; subframe local observations	Beam/column/brace section-index selection	Multi-level code-check and economy rewards	MAPPO
MAFO-3D [7]	3D steel frames; section optimization	Tensorized global state; local observations	Component-group section-index selection	Global and local rewards from magnetic potential differences	MAPPO
MAGIC-Frame [99]	Irregular 3D steel frames; section optimization	Hybrid graph and vector-set state	Component-group section-index selection	Global and local rewards from positive and negative potentials	MAGRPO
Hayashi *et al.* [100]	Planar steel frames; section optimization	Story-wise sections, stresses, and column-overdesign factors (COFs)	Story-wise increase/decrease/keep sections	Feasibility reward; violation zero reward	DQN-assisted metaheuristic search
Hayashi *et al.* [8]	Planar steel frames; section optimization	Graph-embedded member features	Member-wise one-step section update	Feasibility and volume-change rewards	Q-learning
Ororbia *et al.* [41]	Trusses and frames; topology design	Design configuration	Rule-based topology edits	Performance improvement reward	DQN
Ororbia *et al.* [42]	Trusses; topology and parameter design	Graph topology and element parameters	Hierarchical topology/parameter modifications	Objective improvement under constraints	HDRL
Kupwiwat *et al.* [103]	Latticed shells; geometry optimization	Structure and Bézier control-net graphs	Bézier control-point height adjustment	Strain-energy reduction	DDPG
Kupwiwat *et al.* [10]	Trusses; multi-objective optimization	Solution-pool and structure graphs	Section and node-height updates	Hypervolume and Pareto-quality rewards	MADDPG
Li *et al.* [102]	Trusses; topology and geometry optimization	Grid topology and node-coordinate state	Topology actions; coordinate shifts	Deflection, stress, and cost rewards	PPO

Credit assignment is another key issue. A purely global reward makes it difficult to identify the contribution of each local action, whereas purely local rewards may lead agents to improve local indicators at the expense of global performance. Existing studies address this issue through different coordination mechanisms. MAFO-3D combines global and local rewards with value decomposition and a two-level critic architecture [7], whereas FrameMARL coordinates floor-level agents through subframe decomposition, local observations, and global reward feedback [29]. Difference rewards used in multi-objective truss optimization also provide a useful reference, because they evaluate the marginal contribution of local actions to the quality of the solution set [10].

Single-agent RL is suitable for small-scale structural optimization tasks or tasks without a natural decomposition, whereas multi-agent RL is more appropriate for large structures or tasks with naturally distributed design units. However, multi-agent modeling does not eliminate structural coupling. Its practical value depends on reasonable structural decomposition, sufficient global coordination, effective credit assignment, and stable training under code-compliance constraints.

5.5. Evaluation settings and baselines

Evaluation of intelligent structural design and optimization methods should clarify both the comparison reference and the claimed advantage being assessed. Existing studies commonly compare proposed methods with engineer-designed schemes, conventional optimization algorithms, or alternative RL methods. Engineer-designed schemes are especially important in generative structural design, where manual layouts often serve as ground truth labels for model training under the adopted modeling assumptions. Generated layouts are therefore commonly evaluated by their agreement with engineer-designed schemes and further checked through material consumption and structural response analysis [33,61]. For RL-based structural optimization, the evaluation setting should further clarify the baseline, training cost, testing domain, and evaluation criteria. These aspects are compared in Table 3 for representative studies.

For RL-based section optimization, existing studies often adopt simplified or synthetic structural cases because current modeling assumptions are often still insufficient to fully represent practical design scenarios. Such settings can still provide valid algorithmic evidence when different methods are evaluated under consistent problem definitions. Following this principle, Hayashi *et al.* compared RL-guided section optimization with conventional search strategies, including SA and PSO, for discrete section optimization of planar steel frames [8,100]. Fu *et al.* further compared the trained policy with GA-based optimization and exhaustive enumeration under consistent section optimization settings [98].

Comparisons among RL methods under consistent environments are necessary to distinguish the influence of algorithm selection from that of problem formulation. FrameMARL compares its multi-agent policy with a single-agent policy under similar environment and network settings, illustrating the influence of structural decomposition on reward and steel consumption [29]. MAGIC-Frame compares the proposed MAGRPO with a MAPPO baseline. MAPPO showed acceptable training stability but slower learning and weaker improvement in structural-level safety indices [99].

Computational efficiency should also be reported according to the intended use of the trained policy. For single-instance RL, training and deployment should be counted together as the total cost of obtaining a verified optimized design. For reusable policies, training and deployment costs should be reported separately, with clearly defined training and testing domains. Evaluation should therefore include both single-case results, such as feasibility, economy, and convergence, and family-level results on unseen structures. This distinction helps determine whether the reported advantage comes from faster optimization of one case or from policy reuse across structural instances.

Table 3. Comparison of evaluation settings in representative studies.

Study	Baseline	Training cost	Testing domain	Evaluation criteria
FrameRL [98]	Engineer-designed scheme; GA; exhaustive enumeration	300 k episodes; approx. 30 h	Random frame tests; typical/high-rise cases	10 code-based: 7 component (stress/stability/stiffness); 1 joint (strong-column–weak-beam (SCWB)); 2 structural (wind/seismic drift)
FrameMARL [29]	Single-agent RL (FrameSA); GA	300 k episodes; time not reported	Unseen braced-frame benchmarks; 3D validation case	10 code-based: 7 component (stress/stability/stiffness); 1 joint (SCWB); 2 structural (wind/seismic drift)
MAFO-3D [7]	GA	10 k episodes; approx. 53 h	Two irregular 3D archetype cases	11 code-based: 8 component (stress/capacity/stability/deflection); 1 joint (SCWB); 2 structural (wind/seismic drift)
MAGIC-Frame [99]	GA; PSO; MAPPO ablation	20 k episodes; approx. 28 h	Four irregular 3D archetype cases	20 code-based: 8 component (stress/stability/deflection); 1 joint (SCWB); 11 structural (irregularity/modal/drift/torsion/stiffness)
Hayashi *et al*. [100]	SA only; PSO only	1 k episodes; time not reported	8-story training case; 11-story transfer case	3 partly code-based: volume objective; member stress-ratio; joint (COF)
Hayashi *et al*. [8]	PSO	5 k episodes; 25–29 h	Variable-size frame transfer tests	6 partly code-based: volume objective; member stress/deflection; joint COF; structural drift/shear capacity
Ororbia *et al*. [41]	GA; brute force; tabular Q-learning	5 k–50 k episodes; time not reported	Seven case-specific planar examples	3 non-code: displacement objective; stability/volume constraints
Ororbia *et al*. [42]	Discrete/TopOpt reference solutions	30 k–50 k episodes; training time not reported	Four case-specific truss examples	3 non-code: displacement objective; stability/volume constraints
Kupwiwat *et al*. [103]	PSO; SA	1 k episodes; time not reported	Unseen/larger latticed shells	1 non-code: strain-energy objective; no safety checks
Kupwiwat *et al*. [10]	Multi-objective optimization baselines (NSGA-II and MOEA/D)	701 episodes (10-bar); 2001 episodes (bridge/roof); time not reported	10-bar and bridge/roof truss cases	4 non-code: stress; displacement; volume; target-shape objectives
Li *et al*. [102]	Random search; two-stage GA	Stage 1/2: 5 M/50 k steps; time not reported	Randomized truss-bridge cases	5 partly code-based: member stress; structural stability; deflection; volume; zero-force metrics

As shown in Table 3, existing studies report training cost using different indicators, such as episodes, interaction steps, or training time. This makes direct efficiency comparison difficult, especially when RL is positioned as a reusable policy-learning method rather than a single-case optimizer. The comparison also shows that code compliance is usually incorporated through FE-based evaluation and constraint penalties, while final engineering acceptance still requires deterministic verification.

Evaluation of RL-based optimization should also examine the decision behavior of the learned policy. Heuristic optimization mainly searches candidate designs through an objective or fitness function, whereas RL learns a state-conditioned policy from reward-guided interactions. The learned policy is expected to generalize by mapping the structural states of unseen instances to design actions. This capability should be assessed by interpreting policy behavior, including section-adjustment trajectories, action distributions, sensitivity to constraint-margin features, and failure modes. These analyses help explain the policy's decision mechanism and assess whether its decisions reflect engineering-relevant adjustment patterns or artifacts of the reward function or training domain.

6. Challenges and future directions

Despite recent progress, RL-based section optimization remains at an early stage of engineering adoption. Challenges include code-compliance assurance, action space design, computational efficiency, structural-family generalization, multi-agent coordination, and workflow integration, as summarized in Table 4.

Table 4. Challenges and future directions in RL-based optimization of steel structures.

Challenge	Current limitation	Future direction
Code-compliance assurance	Reward penalties are insufficient for strict, auditable code compliance	Integrate code-check feedback, safety filtering, repair, and final verification
Action space design	High action dimensionality and fine-grained grouping hinder exploration	Use hierarchical actions and dynamic filtering based on component roles and constraint states
Computational efficiency	Repeated finite element analyses and code checks make RL training costly	Use multi-fidelity evaluation, physics-aware surrogates, and data-based policy pretraining
Structural-family generalization	Policy robustness beyond limited structural-family domains remains insufficiently validated	Define explicit family domains and test interpolation and extrapolation on unseen structures
Multi-agent coordination	Local agents may become myopic if global performance feedback is not properly assigned	Coordinate local section decisions through credit assignment, information sharing, and reward decomposition
Engineering workflow integration	Existing studies remain weakly connected to design practice and engineering review	Integrate RL with generative design, surrogate evaluation, FE verification, and human–AI collaboration

6.1. Code-compliance assurance

The feasible domain of steel section optimization is governed by strongly coupled and nonsmooth constraints on load-bearing capacity, stability, and detailing, often resulting in a sparse feasible region. Traditional optimization methods commonly handle infeasible designs through penalty, screening, or repair mechanisms [1,5,35]. This issue becomes more pronounced in RL frameworks because the agent must repeatedly explore the boundary of the feasible domain during training. If code constraints are handled only through reward penalties, the learning signal may become unstable, and the policy may even learn an inappropriate trade-off between material reduction and constraint violation.

Existing studies have begun to transform code-checking results into learnable constraint-margin information and to incorporate such information into state representations, reward functions, and action constraints. This allows the agent to perceive the degree of constraint violation or the safety margin of the current design, rather than relying only on a binary feasibility label [7,29,105]. However, these mechanisms should be regarded mainly as feasibility guidance during learning, rather than as compliance guarantees during design. Reward penalties, constraint-margin features, or local action restrictions can encourage the policy to move toward the feasible domain, but they cannot ensure that every generated design strictly satisfies all coupled code provisions, nor can they replace deterministic and auditable engineering verification. For engineering applications, RL-based section optimization should be integrated into a verification workflow in which structural analysis and code-compliance checks evaluate the feasibility of policy-generated designs. During training, constraint information should be incorporated through one or more mechanisms, such as reward functions, action restrictions, safety filters, or repair operators. Final designs must then undergo independent deterministic verification.

6.2. Action space design

Action space design directly affects the trainability and engineering applicability of RL-based optimization. Section optimization of steel frames usually involves a large number of beams, columns, and braces, and different component types are associated with different standard section pools. If sections are directly selected for all components, the action space rapidly expands with the number of components and the size of the section pools, leading to inefficient exploration and increased training difficulty. Component grouping can reduce the dimensionality of the decision space and is also consistent with engineering practice, where components are often grouped by story, component type, or load-resisting characteristics for section assignment [1,5,7,36]. However, overly coarse grouping may compress the searchable design space and reduce local optimization capability, whereas overly fine grouping may reintroduce high-dimensional decision-making and coordination difficulties.

Existing studies have explored action space design for section optimization from different perspectives. For example, studies on planar steel frames often adopt local section adjustment or component-based discrete actions, whereas studies on three-dimensional steel frames tend to organize actions at the component-group or substructure level to reduce the complexity of high-dimensional section decisions [7,8,29]. Nevertheless, systematic comparisons are still lacking regarding how different action parameterizations affect feasible-solution rates, exploration efficiency, final solution quality, and generalization across structural families. Action space design should therefore be driven more by the nature of the structural optimization problem than by the algorithm type alone. A promising direction is to organize section adjustments into hierarchical actions with clearer engineering meaning, and to dynamically filter feasible actions according to component roles, constraint margins, and the overall performance state. Such designs can reduce unproductive exploration while avoiding excessive compression of the design space, thereby balancing search efficiency, engineering applicability, and optimization accuracy.

6.3. Computational efficiency

RL-based optimization of steel structures requires a large number of environment interactions during training, and each interaction may involve finite element analysis, code-compliance checks, and reward evaluation. When high-fidelity analysis is performed at every decision step, computational cost becomes a critical bottleneck. Traditional optimization and surrogate-model-based studies have alleviated similar analysis costs through parallel computing, approximate evaluation, and surrogate modeling [6,13,37]. These strategies provide useful references for improving the training efficiency of RL-based optimization. Otherwise, the large number of analyses required during training may weaken the efficiency advantage of the learned policy during deployment.

Surrogate models can reduce the cost of intermediate evaluations, but their errors near feasible-domain boundaries and in strongly nonlinear response regions may mislead policy learning. A more robust approach is to adopt a tiered evaluation mechanism, in which low-cost models are used to screen most candidate designs, whereas high-fidelity finite element analysis and code-compliance checks are performed for designs close to constraint boundaries or with high optimization potential. The newly obtained results can then be used to update the surrogate model [13,84]. This multi-fidelity strategy helps balance computational efficiency and evaluation reliability.

Physics-informed surrogate models and differentiable analysis frameworks are also worthy of further attention. Physics-informed surrogate models can improve the physical consistency of response prediction by incorporating structural topology, equilibrium relationships, boundary conditions, and constraint margins [92,106]. Differentiable analysis and optimization frameworks may further establish gradient links between design variables and structural responses, providing a basis for combining RL with surrogate-based and gradient-based optimization [93,107]. Another complementary route is to use existing design schemes, traditional optimization trajectories, and finite element analysis samples for policy pretraining, allowing the agent to avoid starting exploration from a fully random policy.

6.4. Structural-family generalization

Structural-family generalization is critical for extending RL from single-case optimization to engineering reuse. Future studies should distinguish different generalization settings within explicitly defined structural-family domains. The first is interpolation within a predefined domain, such as unseen structural instances generated within an MCS-defined range. The second is extrapolation to structures with larger scales, different bracing layouts, loading patterns, or alternative section pools, where the structural configuration still follows the same basic system assumptions but the state distribution and feasible design space may change substantially. Existing studies based on graph representations, standardized environments, and MCS-based domain definitions have provided useful foundations [7,8,28], but the reliability of learned policies under these different generalization settings remains insufficiently validated.

Improving such cross-structure generalization requires state representations that can simultaneously capture component-grouping relationships in regular structural systems and topological variations across different structural scales. A graph–tensor hybrid representation may provide a suitable route. Tensor representations can preserve regularly structured information on stories, component groups, and section labels, whereas graph representations are better suited to describing node–component connectivity and local–global coupling. By incorporating structural context information, this representation may help learned policies adapt to different structural instances. Related studies on graph-based RL and graph-based surrogate models have provided preliminary foundations for this direction [8,18,92].

6.5. Multi-agent coordination

In structural optimization, multi-agent RL is mainly introduced to reduce the decision burden of a single agent in high-dimensional design spaces. By decomposing a large structure into multiple local decision units, each agent can focus on a smaller part of the structure. This decomposition makes the coupling between local and global structural behavior explicit, with local agents operating on decomposed subproblems while their decisions must remain consistent with global structural responses, constraint states, and optimization objectives. Therefore, the structural decomposition strategy and the mechanism for aligning local decisions with global structural responses should be designed jointly.

Future studies should place greater emphasis on credit assignment, information-sharing mechanisms, centralized training with decentralized execution, and reward decomposition strategies. These mechanisms are needed to link local section decisions with global structural performance, while avoiding locally myopic policies or unstable training caused by parallel decision-making. In this sense, multi-agent RL provides a coordination framework for reducing decision dimensionality while accounting for mechanical coupling and engineering-based component grouping in steel structures.

6.6. Integration into the engineering design loop

RL-based section optimization is better positioned within an integrated intelligent structural design loop that combines generative models, RL, surrogate models, and finite element verification. In such a framework, generative models provide candidate structural layouts, RL performs section adjustment and constraint correction, surrogate models reduce evaluation costs, and the final schemes are verified through high-fidelity finite element analysis and code-compliance checks. Related studies on generative design and large-language-model-assisted structural optimization have preliminarily demonstrated the potential of such integrated workflows [33,55,56,108].

Such a closed-loop framework also requires evaluation criteria that go beyond steel weight and mechanical response alone. Traditional optimization studies on steel structures have considered design complexity, life-cycle cost, and seismic loss, while recent intelligent design studies have further extended the evaluation scope to carbon emissions, construction constraints, and detailing requirements [1,3,109]. Future RL-based optimization should therefore include indicators such as section diversity, connection complexity, assembly efficiency, construction cost, and carbon emissions, so that the learned policy supports comprehensive engineering decision-making rather than steel weight minimization alone.

Numerical optimization cannot fully capture the broader requirements of structural engineering design, which also involve code interpretation, constructability assessment, and project-specific design preferences. Therefore, RL should be integrated into a human–AI collaborative design workflow, in which the learned policy provides section-adjustment suggestions, candidate schemes, and constraint-correction strategies, while engineers define design preferences, screen out unreasonable results, and conduct final verification. Such an interactive workflow is essential for transforming RL-based optimization from an automated search technique into an auditable and verifiable engineering design-support tool.

7. Conclusions

This review provides a structured synthesis of existing optimization approaches for steel structures, with particular attention to RL-based section optimization of steel frames. Section optimization should be understood as a highly constrained decision-making process involving discrete section pools, code-compliance constraints, multi-objective trade-offs, and engineering implementation requirements, rather than as a simple minimum-weight search problem.

Traditional optimization and surrogate-assisted methods have provided effective tools for single-case section optimization and computational cost reduction. However, the knowledge gained through these optimizations is usually confined to individual design instances. RL offers a complementary perspective by modeling section adjustment as a sequential decision-making process with state feedback, thereby enabling the learning of reusable adjustment policies across related structural instances.

Existing studies have demonstrated the feasibility of RL for optimizing steel frames, with applications ranging from planar frames to three-dimensional moment-resisting frames and braced frames. These developments have also clarified several key methodological components, including structural representation, environment construction, MCS-based structural-family definition, reward design, and multi-agent decomposition.

These findings suggest that RL should be regarded as a complementary decision-support tool for steel-structure optimization rather than as a standalone substitute for established workflows involving traditional optimization methods, finite element analysis, code-compliance checks, and engineering judgment. Its main value lies in learning reusable section-adjustment policies from repeated design interactions across related structural instances. For engineering applications, such policies should be embedded within an auditable design workflow that includes high-fidelity verification, code-compliance checking, and final professional judgment.

Declaration of generative AI and AI-assisted technologies

During the preparation of this manuscript, the authors used generative AI tools only to improve language and readability. Specifically, the authors used ChatGPT for language polishing only. The authors take full responsibility for the content of the manuscript.

Acknowledgments

This work was supported by the National Key R&D Program of China with Grant No. 2025YFF0519700, the Shanghai Rising-Star Program with Grant No. 24YF2749100, and the National Natural Science Foundation of China (NSFC) with Grant No. 52378182.

Authors' contribution

Conceptualization, Yuqing Gao; methodology, Yuqing Gao and Yuhang Lu; validation, Yuqing Gao and Meiyu Du; formal analysis, Yuqing Gao; investigation, Yuhang Lu and Meiyu Du; resources, Wei Wang; data curation, Yuhang Lu, Meiyu Du and Guanren Zhou; writing—original draft preparation, Yuhang Lu; writing—review and editing, Yuqing Gao, Meiyu Du, Wei Wang and Guanren Zhou; visualization, Yuhang Lu; supervision, Yuqing Gao; project administration, Yuqing Gao; funding acquisition, Yuqing Gao and Wei Wang. All authors have read and agreed to the published version of the manuscript.

Conflicts of interest

The authors declare no conflicts of interest.

References

[1] Liu M, Burns SA, Wen YK. Genetic algorithm based construction-conscious minimum weight design of seismic steel moment-resisting frames. *J. Struct. Eng.* 2006 132(1):50–58.

[2] Gholizadeh S, Fattahi F. Damage-controlled performance-based design optimization of steel moment frames. *Struct. Des. Tall Spec. Build.* 2018 27(14):e1498.

[3] Gholizadeh S, Hasançebi O. Efficient neural network-aided seismic life-cycle cost optimization of steel moment frames. *Comput. Struct.* 2024 301:107443.

[4] Hagishita T, Ohsaki M. Optimal placement of braces for steel frames with semi-rigid joints by scatter search. *Comput. Struct.* 2008 86(21–22):1983–1993.

[5] Wan Y, Hu B, Yang Y, Jin F, Zhou J, *et al.* Discrete sizing optimization method based on dividing rectangles algorithm and local response surface for steel frame structures. *J. Build. Eng.* 2023 79:107826.

[6] Shan W, Liu J, Zhou J. Integrated method for intelligent structural design of steel frames based on optimization and machine learning algorithm. *Eng. Struct.* 2023 284:115980.

[7] Du M, Wang W, Gao Y, Fu B. MAFO-3D: a multi-agent reinforcement learning framework for efficient optimization of 3D steel moment-resisting frames. *Eng. Struct.* 2025 345:121574.

[8] Hayashi K, Ohsaki M. Graph-based reinforcement learning for discrete cross-section optimization of planar steel frames. *Adv. Eng. Inform.* 2022 51:101512.

[9] Li H, Liu J, Li J, Wei W, Shan W. Intelligent optimization method for complex steel structures based on internal force state. *J. Constr. Steel Res.* 2024 218:108732.

[10] Kupwiwat Ct, Hayashi K, Ohsaki M. Multi-objective optimization of truss structure using multi-agent reinforcement learning and graph representation. *Eng. Appl. Artif. Intell.* 2024 129:107594.

[11] de Santana Gomes WJ, Beck AT. Global structural optimization considering expected consequences of failure and using ANN surrogates. *Comput. Struct.* 2013 126:56–68.

[12] Gudipati VK, Cha EJ. A framework for optimization of target reliability index for a building class based on aggregated cost. *Struct. Saf.* 2019 81:101873.

[13] Do B, Ohsaki M. Sequential sampling approach to energy-based multi-objective design optimization of steel frames with correlated random parameters. *Earthq. Eng. Struct. Dyn.* 2022 51(3):588–611.

[14] Mitchell M. *An Introduction to Genetic Algorithms*, Cambridge: MIT Press, 1998.

[15] Eberhart R, Kennedy J. A new optimizer using particle swarm theory. In *Proceedings of the Sixth International Symposium on Micro Machine and Human Science (MHS'95)*, Nagoya, Japan, October 4–6, 1995, pp. 39–43.

[16] Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. *Science* 1983 220(4598):671–680.

[17] Truong VH, Cao TS, Tangaramvong S. A robust machine learning-based framework for handling time-consuming constraints for bi-objective optimization of nonlinear steel structures. *Structures* 2024 62:106226.

[18] Cao H, Li M, Nie L, Xie Y, Kong F. Vertex-based graph neural network classification model considering structural topological features for structural optimization. *Comput. Struct.* 2024 305:107542.

[19] Issa O, Silva-Lopez R, Baker JW, Burton HV. Machine-learning-based optimization framework to support recovery-based design. *Earthq. Eng. Struct. Dyn.* 2023 52(11):3256–3280.

[20] Silver D, Huang A, Maddison CJ, Guez A, Sifre L, *et al.* Mastering the game of Go with deep neural networks and tree search. *Nature* 2016 529(7587):484–489.

[21] Lv S, Li J, Zhao C. Reinforcement learning in spatial public goods games with environmental feedbacks. *Chaos Solitons Fractals* 2025 195:116296.

[22] Chai Y, Gao Y, Lu W, Lu X. VertInspect: learning-based humanoid whole-body control for glass curtain wall inspection. *Adv. Eng. Inform.* 2026 73:104590.

[23] Wang Z, Gao Y, Zhou Y, He J, Peng J, *et al.* Toward next-generation autonomous structural health monitoring and inspection with humanoid robots. *Comput.-Aided Civ. Infrastruct. Eng.* 2026 44:100007.

[24] Zhu M, Wang Y, Pu Z, Hu J, Wang X, *et al.* Safe, efficient, and comfortable velocity control based on reinforcement learning for autonomous driving. *Transp. Res. Part C Emerg. Technol.* 2020 117:102662.

[25] Degrave J, Felici F, Buchli J, Neunert M, Tracey B, *et al.* Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature* 2022 602(7897):414–419.

[26] Hafner D, Pasukonis J, Ba J, Lillicrap T. Mastering diverse control tasks through world models. *Nature* 2025 640(8059):647–653.

[27] Guo D, Yang D, Zhang H, Song J, Wang P, *et al.* DeepSeek-R1 incentivizes reasoning in LLMs through reinforcement learning. *Nature* 2025 645(8081):633–638.

[28] Du M, Gao Y, Wang W, Fu B. FrameGym: a reinforcement learning environments for steel frame structures. *Eng. Struct.* 2025 343:120991.

[29] Fu B, Gao Y, Wang W, Du M. Autonomous component optimization method for steel braced frame structures based on multi-agent and physics-informed deep reinforcement learning. *Adv. Eng. Inform.* 2026 69:103878.

[30] Qin S, Liao W, Huang S, Hu K, Tan Z, *et al.* AIstructure-copilot: assistant for generative AI-driven intelligent design of building structures. *Smart Constr.* 2024 1(1):0001.

[31] Liu Y, Zhou K, Chen Z, Liu J, Liu H. Research development in intelligent generative design methods for steel modular structures. *Smart Constr.* 2026 3(1):0005.

[32] Gao Y, Yang J, Qian H, Mosalam KM. Multiattribute multitask transformer framework for vision-based structural health monitoring. *Comput.-Aided Civ. Infrastruct. Eng.* 2023 38(17):2358–2377.

[33] Fu B, Gao Y, Wang W. Dual generative adversarial networks for automated component layout design of steel frame-brace structures. *Autom. Constr.* 2023 146:104661.

[34] Sarfarazi S, Mascolo I, Modano M, Guarracino F. Application of artificial intelligence to support design and analysis of steel structures. *Metals* 2025 15(4):408.

[35] Balling RJ. Optimal steel frame design by simulated annealing. *J. Struct. Eng.* 1991 117(6):1780–1795.

[36] Jin F, Yang Y, Hu B, Zhou J, Gao B, *et al.* Research on section dimension optimization of high-rise steel–concrete composite buildings based on improved dividing rectangle algorithm and combined response surface model. *Structures* 2023 58:105437.

[37] Adeli H, Kumar S. Concurrent structural optimization on massively parallel supercomputer. *J. Struct. Eng.* 1995 121(11):1588–1597.

[38] Gu Y, Qin S, Liao W, Lu X. Intelligent design of dimensions of reinforced concrete frame structure components using diffusion models. *Comput. Ind.* 2026 175:104428.

[39] Hayashi K, Ohsaki M. Reinforcement learning and graph embedding for binary truss topology optimization under stress and displacement constraints. *Front. Built Environ.* 2020 6:59–74.

[40] Ororbia ME, Warn GP. Design synthesis through a markov decision process and reinforcement learning framework. *J. Comput. Inf. Sci. Eng.* 2022 22(2):21002.

[41] Ororbia ME, Warn GP. Design synthesis of structural systems as a markov decision process solved with deep reinforcement learning. *J. Mech. Des.* 2023 145(6):61701.

[42] Ororbia ME, Warn GP. Discrete structural design synthesis: a hierarchical-inspired deep reinforcement learning approach considering topological and parametric actions. *J. Mech. Des.* 2024 146(9):91707.

[43] Liao W, Lu X, Huang Y, Zheng Z, Lin Y. Automated structural design of shear wall residential buildings using generative adversarial networks. *Autom. Constr.* 2021 132:103931.

[44] Lu X, Liao W, Zhang Y, Huang Y. Intelligent structural design of shear wall residence using physics-enhanced generative adversarial networks. *Earthq. Eng. Struct. Dyn.* 2022 51(7):1657–1676.

[45] Fei Y, Liao W, Zhang S, Yin P, Han B, *et al.* Integrated schematic design method for shear wall structures: a practical application of generative adversarial networks. *Buildings* 2022 12(9):1295.

[46] Goodfellow IJ, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, *et al.* Generative adversarial nets. In *Proceedings of the Advances in Neural Information Processing Systems*, Montreal, Canada, December 8–13, 2014 .

[47] Zhao P, Liao W, Huang Y, Lu X. Intelligent design of shear wall layout based on graph neural networks. *Adv. Eng. Inform.* 2023 55:101886.

[48] Zhao P, Fei Y, Huang Y, Feng Y, Liao W, *et al.* Design-condition-informed shear wall layout design based on graph neural networks. *Adv. Eng. Inform.* 2023 58:102190.

[49] Zhao P, Liao W, Huang Y, Lu X. Beam layout design of shear wall structures based on graph neural networks. *Autom. Constr.* 2024 158:105223.

[50] Zhang C, Tao M, Wang C, Fan J. End-to-end generation of structural topology for complex architectural layouts with graph neural networks. *Comput.-Aided Civ. Infrastruct. Eng.* 2024 39(5):756–775.

[51] Xia J, Liao W, Han B, Zhang S, Lu X. Intelligent co-design of shear wall and beam layouts using a graph neural network. *Autom. Constr.* 2025 172:106024.

[52] Qin S, Liao W, Huang Y, Zhang S, Gu Y, *et al.* Intelligent design for component size generation in reinforced concrete frame structures using heterogeneous graph neural networks. *Autom. Constr.* 2025 171:105967.

[53] Liao W, Huang Y, Zheng Z, Lu X. Intelligent generative structural design method for shear wall building based on "fused-text-image-to-image" generative adversarial networks. *Expert Syst. Appl.* 2022 210:118530.

[54] Gu Y, Huang Y, Liao W, Lu X. Intelligent design of shear wall layout based on diffusion models. *Comput.-Aided Civ. Infrastruct. Eng.* 2024 39(23):3610–3625.

[55] Zhou Y, Leng H, Meng S, Wu H, Zhang Z. StructDiffusion: end-to-end intelligent shear wall structure layout generation and analysis using diffusion model. *Eng. Struct.* 2024 309:118068.

[56] Li J, Wang W, Fu B, Gao Y. FrameDiffusion: a latent diffusion model for intelligent layout design of steel frame-braced structures. *Eng. Struct.* 2025 343:121195.

[57] Fei Y, Liao W, Huang Y, Lu X. Knowledge-enhanced generative adversarial networks for schematic design of framed tube structures. *Autom. Constr.* 2022 144:104619.

[58] Zhao P, Liao W, Xue H, Lu X. Intelligent design method for beam and slab of shear wall structure based on deep learning. *J. Build. Eng.* 2022 57:104838.

[59] Zhao P, Liao W, Huang Y, Lu X. Intelligent design of shear wall layout based on attention-enhanced generative adversarial network. *Eng. Struct.* 2023 274:115170.

[60] Feng Y, Fei Y, Lin Y, Liao W, Lu X. Intelligent generative design for shear wall cross-sectional size using rule-embedded generative adversarial network. *J. Struct. Eng.* 2023 149(11):4023161.

[61] Fu B, Wang W, Gao Y. Physical rule-guided generative adversarial network for automated structural layout design of steel frame-brace structures. *J. Build. Eng.* 2024 86:108943.

[62] Fei Y, Liao W, Lu X, Taciroglu E, Guan H. Semi-supervised learning method incorporating structural optimization for shear-wall structure design using small and long-tailed datasets. *J. Build. Eng.* 2023 79:107873.

[63] Zhang F, Wang K, Sun Q, Song W, Fang C, *et al.* Knowledge-guided generative adversarial networks for intelligent design of internal supporting structures in foundation pits. *Comput.-Aided Civ. Infrastruct. Eng.* 2025 40(30):6149–6164.

[64] He Z, Zhang J, Wang Y. Revisiting the generation framework in generative AI-based intelligent structural design. In *Proceedings of the ASCE International Conference on Computing in Civil Engineering 2025*, 2025, pp. 365–374.

[65] Han J, Lu X, Gu Y, Liao W, Cai Q, *et al.* Optimized data representation and understanding method for the intelligent design of shear wall structures. *Eng. Struct.* 2024 315:118500.

[66] Leng H, Gao Y, Zhou Y. ArchiDiffusion: a novel diffusion model connecting architectural layout generation from sketches to shear wall design. *J. Build. Eng.* 2024 98:111373.

[67] Luo R, Wang Y, Xiao W, Zhao X. AlphaTruss: monte carlo tree search for optimal truss layout design. *Buildings* 2022 12(5):641–662.

[68] Luo R, Wang Y, Liu Z, Xiao W, Zhao X. A reinforcement learning method for layout design of planar and spatial trusses using kernel regression. *Appl. Sci.* 2022 12(16):8227.

[69] Kupwiwat Ct, Hayashi K, Ohsaki M. Deep deterministic policy gradient and graph convolutional network for bracing direction optimization of grid shells. *Front. Built Environ.* 2022 8:899072.

[70] Lin C, Fu B, Zhang L, Li N, Tong GS. Intelligent design of steel–concrete composite beams based on deep reinforcement learning. *Structures* 2024 70:107666.

[71] Lai Y, Ke K, Wang L, Wang L. BIM-based intelligent optimization of complex steel joints using SVM and NSGA-II. *J. Constr. Steel Res.* 2024 223:109086.

[72] Zhang Y, Shen X, Zhang A, Shang-guan G, Chen X. Intelligent design of enveloping tubular staggered flange joints based on GAT. *J. Constr. Steel Res.* 2024 220:108842.

[73] Sarfarazi S, Shamass R, Rabi M, Abarkan I, Ferreira FPV, *et al.* Inverse machine learning for the design of perforated beams: parent section and material prediction. *Eng. Appl. Artif. Intell.* 2026 164:113275.

[74] Fang C, Ping Y, Gao Y, Zheng Y, Chen Y. Machine learning-aided multi-objective optimization of structures with hybrid braces—framework and case study. *Eng. Struct.* 2022 269:114808.

[75] Fadel Miguel LF, Lopez RH, Torii AJ, Beck AT. Reliability-based optimization of multiple folded pendulum TMDs through efficient global optimization. *Eng. Struct.* 2022 266:114524.

[76] Hasançebi O, Carbas S. Bat inspired algorithm for discrete size optimization of steel frames. *Adv. Eng. Softw.* 2014 67:173–185.

[77] Shan W, Zhou X, Liu J, Ding Y, Zhou J. Two-stage automatic structural design of steel frames based on parametric modeling and multi-objective optimization. *Struct. Multidiscip. Optim.* 2024 67(6):104–124.

[78] Park HS, Adeli H. A neural dynamics model for structural optimization—application to plastic design of structures. *Comput. Struct.* 1995 57(3):391–399.

[79] Park HS, Adeli H. Distributed neural dynamics algorithms for optimization of large steel structures. *J. Struct. Eng.* 1997 123(7):880–888.

[80] Park HS, Adeli H. Data parallel neural dynamics model for integrated design of large steel structures. *Comput.-Aided Civ. Infrastruct. Eng.* 1997 12(5):311–326.

[81] Sarfarazi S, Modano M, Fulgione M. Artificial intelligence for dynamic characterization of composite panel structures: a structured review. *Mech. Res. Commun.* 2026 151:104607.

[82] Motamedi P, Banazadeh M, Talatahari S. Seismic loss optimum design of steel structures using learning-based charged system search. *Struct. Des. Tall Spec. Build.* 2022 31(13):e1945.

[83] Xiao Y, Yue F, Zhang X, Shahzad MM. Aseismic optimization of mega-sub controlled structures based on gaussian process surrogate model. *KSCE J. Civ. Eng.* 2022 26(5):2246–2258.

[84] Peng C, Guo T, Chen C, Xu W. A real-time hybrid simulation framework for reliability-based design optimization of structures subjected to pulse-like ground motions. *Earthq. Eng. Struct. Dyn.* 2024 53(10):3246–3262.

[85] Wang Z, Mulyanto JA, Zheng C, Wu Y. Research on a surrogate model updating-based efficient multi-objective optimization framework for supertall buildings. *J. Build. Eng.* 2023 72:106702.

[86] Huang M, Wang C, Lin W, Xiao Z. A multi-objective structural optimization method for serviceability design of tall buildings. *Struct. Des. Tall Spec. Build.* 2023 32(17):e2052.

[87] Cao H, Li H, Sun W, Xie Y, Huang B. A boundary identification approach for the feasible space of structural optimization using a virtual sampling technique-based support vector machine. *Comput. Struct.* 2023 287:107118.

[88] Prayogo H, Yang IT, Cheng MY. Dynamic gradient boosted metaheuristic approach for efficient reinforced concrete structure optimization. *J. Build. Eng.* 2024 97:110864.

[89] Li Y, Liu Y, Yu H, Ma K, Zhang X, *et al.* Multi-objective optimization design of coupled wall structure with hybrid coupling beams using hybrid machine learning algorithms. *J. Build. Eng.* 2023 78:107745.

[90] Li Z, Yang Q, Deng Q, Gong Y, Tian D, *et al.* Time history seismic response prediction of multiple homogeneous building structures using only one deep learning-based structure temporal fusion network. *Earthq. Eng. Struct. Dyn.* 2024 53(13):4076–4098.

[91] Xia Y, Liu J, Qi H. A population-based DNN-augmented optimization method for designing truss structures. *Swarm Evol. Comput.* 2024 89:101613.

[92] Kuo PC, Chou YT, Li KY, Chang WT, Huang YN, *et al.* GNN-LSTM-based fusion model for structural dynamic responses prediction. *Eng. Struct.* 2024 306:117733.

[93] Zhang C, Tao M, Wang C, Yang C, Fan J. Differentiable automatic structural optimization using graph deep learning. *Adv. Eng. Inform.* 2024 60:102363.

[94] Fei Y, Qin S, Liao W, Guan H, Lu X. Graph neural network-assisted evolutionary algorithm for rapid optimization design of shear-wall structures. *Adv. Eng. Inform.* 2025 65:103129.

[95] Liu R, Ma G, Xie X, Qu T, Liu B, *et al.* Agent of deep reinforcement learning for multi-objective arch dam shape design. *Comput.-Aided Civ. Infrastruct. Eng.* 2025 40(27):4970–4993.

[96] Li S, Snaiki R, Wu T. A knowledge-enhanced deep reinforcement learning-based shape optimizer for aerodynamic mitigation of wind-sensitive structures. *Comput.-Aided Civ. Infrastruct. Eng.* 2021 36(6):733–746.

[97] Zhu S, Ohsaki M, Hayashi K, Guo X. Machine-specified ground structures for topology optimization of binary trusses using graph embedding policy network. *Adv. Eng. Softw.* 2021 159:103032.

[98] Fu B, Gao Y, Wang W. A physics-informed deep reinforcement learning framework for autonomous steel frame structure design. *Comput.-Aided Civ. Infrastruct. Eng.* 2024 39(20):3125–3144.

[99] Du M, Wang W, Gao Y. MAGIC-frame: multi-agent GRPO-based intelligent code-compliant optimization for irregular steel frame structures. *Eng. Struct.* 2026 366:123305.

[100] Hayashi K, Ohsaki M. Reinforcement learning for optimum design of a plane frame under static loads. *Eng. Comput.* 2021 37(3):1999–2011.

[101] Kakooee R, Dillenburger B. Reimagining space layout design through deep reinforcement learning. *J. Comput. Des. Eng.* 2024 11(3):43–55.

[102] Li M, Zheng Q, Ashuri B. Application of artificial intelligence in design automation: a two-stage framework for structure configuration and design. *J. Constr. Eng. Manag.* 2024 150(8):4024083.

[103] Kupwiwat Ct, Hayashi K, Ohsaki M. Deep deterministic policy gradient and graph attention network for geometry optimization of latticed shells. *Appl. Intell.* 2023 53(17):19809–19826.

[104] Liu J, Liu P, Feng L, Wu W, Li D, *et al.* Automated clash resolution for reinforcement steel design in concrete frames via Q-learning and building information modeling. *Autom. Constr.* 2020 112:103062.

[105] Li J, Gao Y, Wang W, Lian Z, Fu B. Deep reinforcement learning-based design optimization method for suspended ceiling systems of space grid structures. *Adv. Eng. Inform.* 2026 72:104482.

[106] Fei Y, Liao W, Zhao P, Lu X, Guan H. Hybrid surrogate model combining physics and data for seismic drift estimation of shear-wall structures. *Earthq. Eng. Struct. Dyn.* 2024 53(10):3093–3112.

[107] Song LH, Wang C, Fan JS, Lu HM. Elastic structural analysis based on graph neural network without labeled data. *Comput.-Aided Civ. Infrastruct. Eng.* 2023 38(10):1307–1323.

[108] Qin S, Guan H, Liao W, Gu Y, Zheng Z, *et al.* Intelligent design and optimization system for shear wall structures based on large language models and generative artificial intelligence. *J. Build. Eng.* 2024 95:109996.

[109] Hosseinzadeh A, Dehestani M. Intelligent low carbon reinforced concrete beam design optimization via deep reinforcement learning. *Sci. Rep.* 2025 15(1):33143.