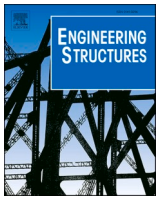
Engineering Structures 353 (2026) 122272

Contents lists available at ScienceDirect

Engineering Structures

journal homepage: www.elsevier.com/locate/engstruct



Automated design of shear wall structures using reinforcement learning with code compliance

Guoxu Zhao [a], Gao Ma [a,b,*], Min-Jeong Cho [c], Hyeon-Jong Hwang [c,**]

[a] College of Civil Engineering, Hunan University, Changsha 410082, China
[b] National Key Laboratory of Bridge Safety and Resilience, Hunan University, Changsha 410082, China
[c] Department of Architecture and Architectural Engineering, Seoul National University, 1 Gwanak-ro, Seoul 08826, Republic of Korea

ARTICLE INFO

Keywords:
Deep reinforcement learning
Automation design
Shear wall
Soft actor-critic

ABSTRACT

With the advancement of artificial intelligence technologies, automated structural design has emerged as a new research focus in recent years. This study proposes a deep reinforcement learning (DRL) method for design of reinforced concrete shear walls. Based on the Soft Actor-Critic (SAC) algorithm, DRL agent was trained with deep learning network. Design code-based constraints were incorporated into the DRL framework to automate structural design while ensuring compliance with design codes. Additionally, for structures where the shear strength cannot be accurately determined, a machine learning technique for shear strength estimation was integrated with DRL on automated design. For DRL training, reward functions were established to meet the design code requirements and minimize cost. After 100,000 training iterations, DRL agent was successfully trained to design shear walls with an acceptable cost. This method is capable of designing compliant and cost-efficient shear wall structures within only 0.1 s. Further, a practical case study was conducted to compare DRL with traditional optimization algorithms, demonstrating the superior computational efficiency of DRL in automatic design of shear walls. The speed and accuracy of the DRL design agent validated the feasibility of the proposed method, and highlighted its potential in effectively solving complex civil engineering design problems.

1. Introduction

Reinforced concrete shear wall structures have been widely used in building structures, but the design process is typically iterative, involving the determination of multiple design variables, such as material properties, geometric parameters, and layout configurations. Conventional design methods often rely on engineers' experience and a trial-and-error process, making it challenging to identify an optimal shear wall structure in a short time [1]. To address this issue, it is essential to develop intelligent design technologies centered on parameterization and automation [2].

In the early stages of artificial intelligence (AI) applications in structural design, expert systems were primarily used to simulate the reasoning and decision-making processes of human experts [3]. Rafig et al. [4] introduced the application of artificial neural networks (ANNs) in engineering and utilized ANNs for the design of prestressed reinforced concrete slabs. Tabarak et al. [5] explored the use of ANNs for selecting appropriate structural systems. Jootoo et al. [6] applied machine learning algorithms to preliminary bridge design problems, leveraging machine learning for bridge type selection. Preliminary explorations of structural topology optimization methods based on deep learning and reinforcement learning were conducted, enabling the direct generation of optimized structural topologies under given external loads [7–9]. Zhang et al. [10] developed a system that employed an improved particle swarm optimization (PSO) algorithm to optimize shear wall layout, addressing the structural performance, architectural configuration, and structural weight reduction. Similarly, Lou et al. [11] adopted an improved PSO algorithm for optimizing shear wall structures and proposed a novel method for estimating the corresponding structural responses. Their approach improved the efficiency of structural optimization and reduced material consumption. Atabay [12] used genetic algorithms to optimize shear wall structures with the aim of identifying cost-minimizing design. Although these structural design optimization methods [4–12] are suitable for one-time structural design, where each structure requires separate computational optimization, they enhance design efficiency to a certain extent. In recent years,

* Corresponding author at: College of Civil Engineering, Hunan University, Changsha 410082, China.
** Corresponding author.
E-mail addresses: zhaoguooxu@hnu.edu.cn (G. Zhao), magao@hnu.edu.cn (G. Ma), fjgh1998@snu.ac.kr (M.-J. Cho), hwanghj@snu.ac.kr (H.-J. Hwang).

https://doi.org/10.1016/j.engstruct.2026.122272
Received 2 September 2025; Received in revised form 4 January 2026; Accepted 1 February 2026

numerous researchers have applied optimization and learning algorithms [13,14] to the automated design of shear wall structures. Due to the powerful feature extraction capabilities of deep neural networks (DNNs), the integration of generative adversarial networks (GANs) [15–17] has become a mainstream trend in the field of automated structural design, focusing on the layout design of structures. Lu et al. [18,19] utilized GANs to generate layout for wall structures. However, this approach relies heavily on prior knowledge and requires a substantial dataset of engineering design drawings for training. Unfortunately, access to such datasets is often limited, as most design drawings are held by design institutes, making data collection challenging. The application of large language models (LLMs) in structural design also represents a form of data-free design [20,21]. For instance, Chen and Bao [20] employed an LLM to generate code-compliant reinforced concrete designs and verified the results using SAP2000, demonstrating the feasibility of LLM-based structural design. However, the primary limitations of LLMs in this field lie in their questionable structural rigor, limited physical interpretability, and potential unreliability in ensuring content accuracy.

In structural design, if structural components have nonlinear behavior in structure analysis, optimization algorithms would become trapped in local optima, making it difficult to find the global optimum [22]. Moreover, for complex structural design problems, these algorithms often face challenges related to computational complexity and convergence speed, especially in high-dimensional and multi-variable scenarios. Additionally, such automated design methods using GAN are currently limited to shear wall layout, with little research focusing on structural components (e.g., RC shear wall section design).

Deep reinforcement learning (DRL) offers a viable option for achieving automated shear wall design effectively, In the absence of sufficient data, automated structural design can be conducted using code-based analytical formulations or finite element analyses. When reliable datasets are available, machine learning techniques can be incorporated to rapidly predict structural load-carrying capacity, thereby significantly reducing computational time. In recent years, DRL has demonstrated tremendous potential in various fields, such as protein design [23,24], mechanical engineering design [25], Atari games [26], controller tuning [27], and self-driving cars [28]. These applications are typically characterized by a large number of design parameters and high-dimensional decision spaces, where traditional metaheuristic optimization algorithms often suffer from low efficiency or premature convergence. DRL offers a fundamentally different solution paradigm for such complex problems and provides a promising new direction for achieving fully automated structural design. Fig. 1 shows an overview of the DRL, which is a hybrid approach that combines DL techniques with reinforcement learning (RL) principles [26,29–31]. The standard RL framework consists of three key components: agent, environment, and their interactions (including actions, states, and rewards). The agent interacts with the environment by taking actions and receiving rewards, then adjusts its action policy to achieve the highest rewards. By simulating the exploratory learning and feedback optimization of an agent within an environment, DRL continuously updates design strategies, enabling complex multi-objective optimization problems to be solved more effectively through automation.

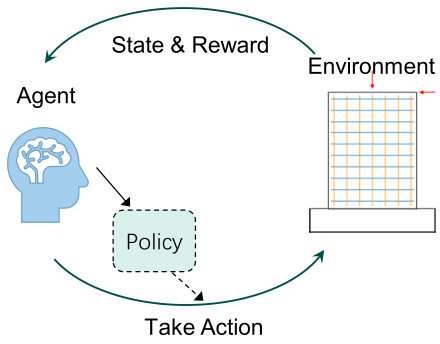


Fig. 1. Overview of deep reinforcement learning.

In civil engineering field, there only exist very limited studies and explorations focusing on DRL for structural design: shear wall layout design [32]; RC beam design [33]; cross-section optimization of steel frame [34,35]; and framework design [36]. Notably, these studies [32–36] primarily focus on layout design and components that are simple and numerically solvable. The application of DRL-based methods in structural design research faces several challenges, primarily due to the following difficulties in a shear wall design example: a) Structural design involves various materials, and the complexity of shear wall structures makes the calculation of shear strength challenging. This necessitates the use of new methods for capacity estimation; b) DRL-based shear wall design requires the development of a new environment to simulate the design process, ensuring that the iterative process meets the design code requirements and minimizes the material costs; and c) In structural design, it is essential to design the effective and physically meaningful reward functions for the constraints imposed by current design code requirements. Additionally, during the structural design process, finite element analysis is required to estimate the structural behavior of complex or special structures. This approach is constrained by current computational power, and training involving large-scale datasets (i.e., hundreds of thousands of iterations in DRL) would be extremely time-consuming, making it unsuitable at present. In recent years, significant advancements in machine learning have led to exceptional feature extraction capabilities, providing new approaches to DRL. Gradient Boosting model implemented via XGBoost, as a machine learning algorithm, can quickly predict shear strength based on empirical data. In this study, the XGBoost was adopted for surrogate modeling due to its strong generalization capability, fast training and inference speed, and built-in interpretability through feature importance analysis [37,38], and this method is particularly suitable for engineering companies with a large amount of design case data.

In structural design problems, the relationship between the environmental state variables (i.e., structural design parameters and performance) and compliance with design code requirements as well as material costs is complicated. This study aims to develop a DRL-based model for shear wall structures to automate the design process and optimize material costs. The Soft Actor-Critic (SAC) algorithm is integrated with XGBoost to enable automated design and performance prediction of shear wall structures. In addition, design code-based methods for calculating load-carrying capacity are also employed to ensure compliance with design codes, particularly in cases where sufficient data are not available. This approach demonstrates the applicability of DRL in engineering design as an alternative method for design automation. An RC shear wall design environment is created within the DRL framework, where an agent can determine design parameters, such as shear wall dimensions, size of distribution rebars, and the number of rebars, based on a given design code.

2. Concept of Soft Actor-Critic (SAC)

DRL algorithms work by continuously trying to find the optimal policy to achieve a given objective. The agent interacts (i.e., taking actions) with the environment, receiving rewards for its current actions, and adjusts its policy to maximize the reward. For instance, Soft Actor-Critic (SAC), one of the DRL algorithms, not only aims to achieve the highest reward but also requires entropy maximization. This enhances the exploration capabilities of the agent, effectively preventing it from becoming stuck in local optima. SAC uses neural network to approximate the reward after an action is taken, which can deal with more complex state space and action space.

One of the advantages of SAC [39] over other reinforcement learning algorithms, such as deep deterministic policy gradient (DDPG) and deep Q-learning algorithms, is that it is designed for the maximum entropy

reinforcement learning. SAC stands out among most model-free (off--policy) reinforcement learning algorithms due to its excellent convergence properties. Model-free methods learn policies directly through interaction without constructing an explicit environmental model. Also, SAC exhibits high efficiency and stability in autonomous learning, even in cases of low sample efficiency [40]. SAC learns directly from interaction with the environment, enabling stronger exploration and better performance under multimodal rewards. Additionally, the algorithm is more robust, which is critical for optimizing complex structures with many parameters and high-dimensional corresponding spaces. The goal of SAC is to maximize the expected accumulated reward and the entropy of each output action of the policy. The maximum entropy-based policy can be defined as follows:

$$\pi^* = \arg\max_{\pi} \mathrm{E}_{\tau\sim\pi}\left[\sum_{t=0}^{T}\gamma^t(R(s_t,a_t)+\alpha H(\pi(\cdot|s_t)))\right] \tag{1}$$

where $\arg\max_{\pi} \mathrm{E}_{\tau\sim\pi}$ represents the finding a policy π that maximizes the expected value; τ denotes the trajectory generated by the interaction between the agent and environment; γ is the reduction factor; $R(s_t,a_t)$ refers to the action reward obtained by taking action a_t in state s_t; t is the time step; α is a hyper-parameter that represents the importance of entropy value. In the present study, α is automatically learned and treated as a trainable parameter. It is dynamically adjusted through gradient descent to satisfy a predefined entropy target; and $H(\pi(\cdot|s_t))$ is the entropy of π at state s_t.

In the SAC framework, the state value and action-value functions are updated iteratively to guide the policy optimization process, as expressed in Eqs. (2) and (3). The state value function $V(s_t)$ can be described as follows:

$$V(s_t) = \mathbb{E}_{a_t\sim\pi}\left[Q_\theta(s_t,a_t) - \alpha\log\pi_\phi(a_t|s_t)\right] \tag{2}$$

where $\mathbb{E}$ represents an average over randomness; $Q_\theta(s_t,a_t)$ is the action reward function; and $\pi_\phi(a_t|s_t)$ is the policy. The reward function can be expressed as follows:

$$Q_\theta(s_t,a_t) = r(s_t,a_t) + \gamma\mathbb{E}_{s_{t+1}}[V(s_{t+1})] \tag{3}$$

where r is the immediate reward after taking action; and $\gamma\mathbb{E}_{s_{t+1}}[V(s_{t+1})]$ is the expectation over actions chosen by the current policy in the next state.

During the training process, SAC minimizes the Q-value loss functions and policy loss functions to obtain the optimal policy. The Q-value loss function can be defined as follows :

$$J_Q(\theta) = \mathop{\mathbb{E}}_{(s_t,a_t)\sim D}\left[\frac{1}{2}(Q_\theta(s_t,a_t) - (r+\gamma\mathbb{E}_{s'}V(s_{t+1})))^2\right] \tag{4}$$

The loss function for the actor-networks at training the policy π_ϕ can be defined as follows:

$$J_\pi(\phi) = \mathop{\mathbb{E}}_{s_t\sim D, a_t\sim\pi_\phi}\left[\log\pi_\phi(a_t|s_t) - \frac{1}{\alpha}Q_\theta(s_t,a_t) + \log Z(s_t)\right] \tag{5}$$

where $Z(s_t)$ represents the normalization constant under the state s_t.

In a word, the agent continuously generates actions based on a policy to obtain higher rewards, while simultaneously learning from the accumulated experience, which comprises actions, rewards, and states, to improve its policy over time.

3. Methodology

In reinforcement learning, the agent interacts with the environment by continuously taking actions, which causes changes in the environment's state. A typical schematic of RC shear wall for automatic design is shown in Fig. 2. The main parameters include wall height, width, thickness, transverse and longitudinal reinforcements, and axial load conditions, which together define the state and action spaces for the optimization framework.

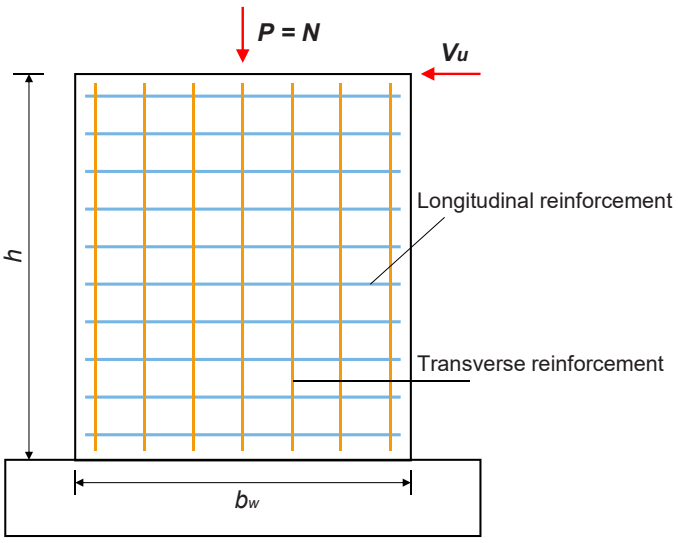


Fig. 2. Typical schematic of the reinforced concrete shear wall.

The goal of DRL is to find a policy of actions that maximizes the cumulative reward. The process of SAC algorithm can be divided into two stages: 1) Exploration and data collection: The agent takes actions in the environment, changes the current state, and receives the corresponding reward values. These interaction data (i.e., state, action, reward, and next state) are recorded in the experience replay buffer for subsequent training. This stage helps the agent accumulate a rich set of sample data, covering feedback information across different states; and 2) Policy optimization and update: SAC uses the collected data to train neural networks, continuously updating the parameters of the policy network and value network. By maximizing the soft Q-value (Eq. (3)), the agent is able to output actions that maximize the reward in future states while maintaining policy diversity. This process enables the agent to gradually approach the optimal policy, where it can select an action that maximizes the reward as well as maintains appropriate exploration in any given state. The design process of shear wall using SAC is illustrated in Fig. 3.

In this study, two approaches are integrated with DRL for automated design of shear wall structures. Method 1 involves directly incorporating existing code-based formulas and constraints to guide the automated design process. This approach does not require the collection of additional data, making it more straightforward for practical implementation. However, in engineering practice, there are some types of structures for which no code-based formula is available to calculate the load capacity, although numerous design cases exist. Therefore, Method 2, which combines XGBoost with DRL, is adopted to account for the influence of factors such as vertical reinforcement and axial load on the shear capacity of shear walls. There are some factors (e.g., longitude reinforcement ratio, axial fore) are often neglected in existing design codes [36–38]. Due to the limited availability of detailed engineering design data for shear walls, it is challenging to directly compare the outcomes of the automated design with actual engineering projects. Thus, this study utilizes a dataset of 774 existing shear wall test specimens, and the design parameter space in the DRL-based model is constrained within the bounds of this dataset. If sufficient engineering design data become available in the future, they could be used to replace or supplement the experimental dataset.

3.1. Environment and state

To achieve automatic design of RC shear walls, a single-story shear

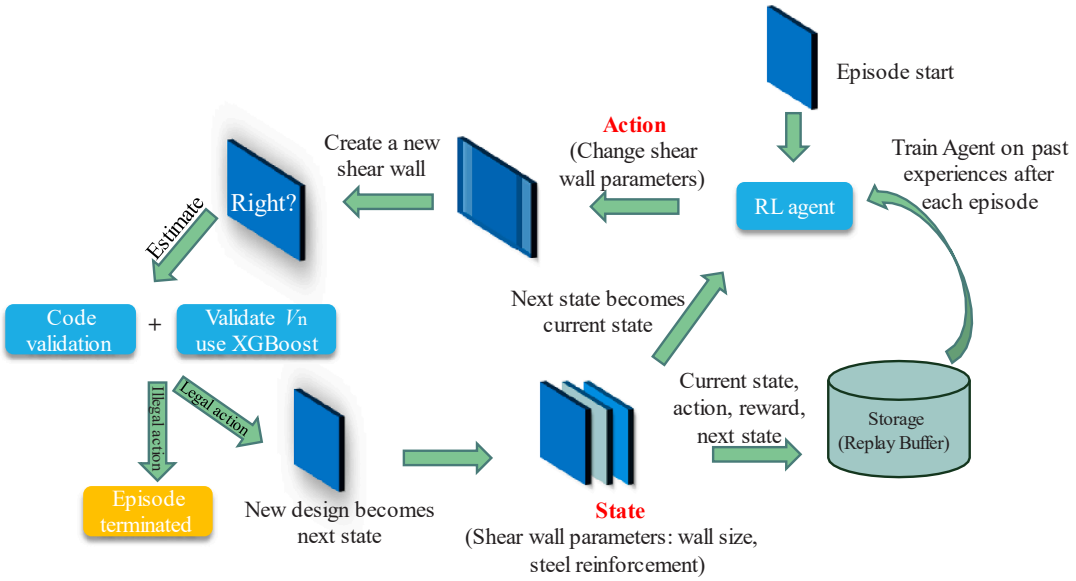


Fig. 3. Flow chart of the design process for optimizing the shear wall structure.

wall is selected for dimensional optimization while minimizing material use. This study focuses solely on conventional RC shear walls, excluding the design of top beams and boundary elements, ensuring compliance with design codes.

The state describes the mechanical and structural information of the shear wall. Based on the state parameters, the current state and configuration of the shear wall can be described. In the computation process, the shear wall's height (h), width (b_w), thickness (t), longitudinal and transverse reinforcement diameter (d_l and d_h) and quantity (N_l and N_h), shear strength (V_n), and shear demand (V_u) are set as the parameters of the environment's state.

It is necessary to define the initial shear wall's conditions (i.e., initial state) for generating a reasonable shear wall structure, such as the shear demand (V_u), wall height (h), seismic grade of the building, concrete compressive strength (f_c), and the strength, number, and size of reinforcement. In this study, the range of these parameters is provided in Table 1, and the used rebar list is shown in Table 2. To reduce training time, certain parameters were set as constant, such as shear wall height, seismic design category, concrete strength, and reinforcement strength. The height of the shear wall is set to 1800 mm, which corresponds to 3600 mm in the real world, considering that the shear wall exhibits symmetric loading and boundary conditions along the vertical direction. The broader the range, the stronger the model's adaptability and generalization ability. However, a wider range would increase the training time of the model. Through DRL training, a well-defined policy can be learned, enabling the model to generate reasonable shear wall structures within the specified parameter ranges.

3.2. Action

In the context of DRL, the agent's task is to design the shear wall's cross-section and reinforcement configuration based on a given shear demand. Actions perform incremental steps to explore in space. In the proposed DRL-based shear wall design framework, the action space is defined to represent incremental modifications of the shear wall's geometric dimensions and reinforcement configuration at each decision step. The action is formulated as a continuous vector:

$$A_{t-\text{norm}} = [a_1, a_2, \ldots, a_n] \in [-1, 1]^n \tag{6}$$

where each component (a_i) corresponds to a normalized control variable associated with a specific design parameter at step (t).

Specifically, the actual meaning of the action vector in this study is defined as follows:

$$\boldsymbol{A}_t = [\Delta b,\ \Delta t,\ \Delta d_h,\ \Delta d_v,\ \Delta n_h,\ \Delta n_v] \tag{7}$$

where Δb is the incremental adjustment of shear wall width; $\Delta t$is the incremental adjustment of shear wall thickness; Δd_h and Δd_v are the incremental adjustment of horizontal and vertical(longitudinal) reinforcement bar diameter (index), respectively; and Δn_h and Δn_v are the incremental adjustment of the number of horizontal and vertical(longitudinal) reinforcement bars, respectively.

Each action component represents a relative adjustment step rather than an absolute design value. The actual physical update of a design variable (x_i) is defined as follows:

$$x_i^{(t+1)} = x_i^{(t)} + \alpha_i a_i \tag{8}$$

Table 1
Setting of design parameters in the environment.

Variable	Unit	Range
Shear demand (V_u)	kN	200–1500
Wall height (h)	mm	1800
Seismic grade	-	III
Compressive strength (f_c)	MPa	30
Elastic modulus (E_c)	MPa	30000
Rebar diameter (d_b)	mm	6–28
Rebar strength (f_y)	MPa	400

Table 2
Rebar diameter and cross-sectional area.

Index	d_b (mm)	A_s (mm^2)
1	6	28.3
2	8	50.3
3	10	78.5
4	12	113.1
5	14	153.9
6	16	201.1
7	18	254.5
8	20	314.2
9	22	383
10	25	490.9
11	28	615.8

Note: d_b is the rebar diameter; and A_s is the cross-sectional area.

where (α_i) is a predefined scaling factor that maps the normalized action (a_i in [-1,1]) to a physically meaningful engineering increment. The values of α_i are set as [500, 50, 3, 3, 10, 10].

To ensure numerical stability and bounded design updates, all action outputs from the Soft Actor-Critic (SAC) policy network are passed through a hyperbolic tangent (Tanh) activation function, which constrains the action space to ([-1,1]). This normalization prevents excessively large design changes within a single step and promotes smooth convergence during training. Within each episode, the agent is allowed to perform a fixed number of action steps (10 steps in this study) to progressively refine the shear wall design. Through successive incremental updates, the agent seeks an optimal balance between structural safety constraints and cost efficiency while satisfying the prescribed shear demand.

3.3. Constraints

Based on the design code (GB50011–2010) [41] and technical specification for concrete structures of tall building (JGJ 3–2010) [42], an integrated DRL environment for shear strength prediction and structural design was established. A typical schematic of RC shear wall for automatic design is shown in Fig. 2. The following requirements should be typically satisfied for automated structural design.

3.3.1. Shear strength requirements

In this study, the design shear value applied in the DRL framework has already incorporated the effects of various load combinations according to the relevant design codes. Therefore, the combined shear demand was directly used for structural design. To evaluate the shear strength of shear walls, two different approaches are adopted in this study for comparison. The first approach (Method 1) follows the code-based analytical formulas derived from existing design standards, while the second approach (Method 2) utilizes a data-driven prediction method based on XGBoost system. The comparison between these two methods allows for assessing the applicability and accuracy of machine learning in predicting shear strength relative to conventional code-based methods.

3.3.1.1. Method 1. Shear demand V_u of shear wall is defined as follows:

$$V_u = \eta_{vw} V_w \tag{9}$$

where V_w is the design shear force for load combinations including earthquake effects; and η_{vw} is the shear amplification coefficient (= 1.6, 1.4, and 1.2 for grades I, II and III, respectively).

In the calculation of the shear strength of shear walls, Chinese design code [41,42] considers only the contribution of concrete, while the reinforcement is required merely to satisfy construction requirements rather than to contribute explicitly to shear strength. Under permanent and temporary design conditions, the dimensions and shear strength of the shear wall should satisfy Eqs. (10) and (11).

$$V_u \le 0.25\beta_c f_c b_w t \tag{10}$$

$$V_u \le \frac{1}{\lambda - 0.5}\left(0.5 f_t b_w h + 0.13 N \frac{A_w}{A}\right) + f_{yh}\frac{A_{sh}}{s} h \tag{11}$$

where β_c is the concrete strength influence factor (β_c= 1 for concrete strength $f_c \le 50$ MPa, and $\beta_c = 0.8$ for $f_c > 80$ MPa); β_c is linearly interpolated for f_c = 50–80 MPa; b_w is the shear wall width; and t is the shear wall thickness; λ is the shear-span ratio which ranges between 1.5 and 2.2; f_t is the tensile strength of concrete; N is the axial load; A_w and A are the web area and total cross-sectional area of the shear wall; and f_{yh}, A_{sh}, and s denote the yield strength, total cross-sectional area, and spacing of transverse reinforcement, respectively.

When seismic design conditions are considered, the shear strength is defined as follows:

$$V_u \le \frac{1}{\gamma_{RE}}(\alpha_c \beta_c f_c b_w t) \tag{12}$$

where α_c is the coefficient related to the shear-span ratio λ ($\alpha_c = 0.2$ for λ>2.5, $\alpha_c = 0.15$ otherwise); and γ_{RE} is the seismic adjustment factor (γ_{RE}= 0.85).

3.3.1.2. Method 2. For seismic resistant design of wall structures, it is crucial to accurately calculate the shear strength of shear walls. Conventional formulas specified in design codes simplify computational procedures for many complex structures, which often yields highly variable and imprecise estimates. ACI 318–19 [43] does not account for the contribution of longitudinal reinforcement to the shear strength, while Hwang et al. [44] argue the significant effect of longitudinal reinforcement on the shear strength. Although finite element analysis (FEA) methods provide higher accuracy results, it requires extensive computational time, making it inefficient for DRL, as numerous iterative steps are needed even when simplifying the FEA models. Considering current computational limitations, FEA methods are unsuitable for training DRL models. Therefore, using machine learning to consider multiple factors to predict shear strength is a simple and effective approach. A wide range of machine learning methods are available for prediction tasks, and many of them have demonstrated satisfactory performance in regression problems. Among these methods, XGBoost [45] has been extensively validated in previous studies and shown to be particularly effective and reliable for applications similar to the one considered in this work [46,47]. Therefore, XGBoost is adopted in this study to establish the nonlinear mapping between shear wall design variables and corresponding shear strength. XGBoost is an optimized distributed gradient boosting system, which incrementally constructs a set of weak learners by minimizing a regularized objective function that balances prediction accuracy and model complexity. Owing to its strong capability in handling high-dimensional input spaces, capturing complex nonlinear interactions, and maintaining robustness against overfitting, XGBoost has been widely validated in structural engineering applications for response prediction [48,49].

Specifically, XGBoost in this study is employed to compute the shear strength based on geometric and reinforcement parameters, while other structural checks, including stability, reinforcement ratio, and axial compression ratio, are conducted in accordance with the relevant design codes. This hybrid strategy allows the DRL agent to optimize the design decisions efficiently while ensuring compliance with established engineering standards. The workflow of the integrated XGBoos-DRL framework is illustrated in Fig. 4.

Feng et al. [47] utilized the XGBoost to predict the shear capacity of RC shear walls, using a database comprising 434 specimens. In this study, the model was adopted to predict the shear strength of shear walls, with the database expanded using the ACI 445B Shear Wall

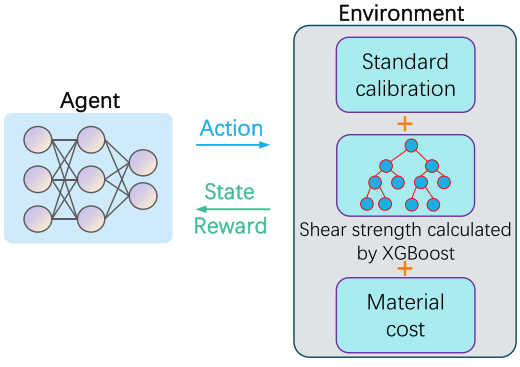


Fig. 4. Flowchart of XGBoost in DRL.

Database [50], resulting in a total of 774 test data. The dataset was randomly divided into a training set (80 %) and a testing set (20 %) [38] to evaluate the generalization performance of the XGBoost. The detailed input variables are the height h, width b_w, web thickness t, concrete compressive strength f_c, longitude reinforcement ratio ρ_v, transverse reinforcement ratio ρ_h and strength fy, and applied axial load P. The prediction accuracy of the model was re-validated. As seen in Fig. 5, XGBoost provided accurate shear strength predictions, and the test data with a mean (A) and overall absolute error (E) of 1.017 and 0.095, respectively. To ensure sufficient safety performance in the actual structural design process, a 95 % one-sided confidence bound [51] should be applied, resulting in a model margin of approximately 19.6 %. Therefore, in the present study we apply a conservative reduction to the XGBoost predicted shear strength and use $V = V_{\text{pred}}(1-0.196)$ (i.e., $\approx 0.80 \cdot S_{\text{pred}}$) when evaluating designs inside the DRL reward. Nevertheless, the final design must still comply with code-based checks for stability, reinforcement limits, and axial compression ratio, ensuring conservative and safe structural performance.

3.3.2. Cross-section dimensions of shear walls

In this study, a straight-shaped configuration is adopted for the shear wall. On the basis of the dataset, the minimum wall thickness is set to 100 mm.

$$t_{\min} = 100 \tag{13}$$

3.3.3. Reinforcement requirements

In high-rise shear wall structures, longitudinal and transverse reinforcement should not be arranged in a single row. For the shear wall with a thickness of 400 mm or less, double-row reinforcement can be used. For the shear wall with a thickness greater than 400 mm but not exceeding 700 mm, triple-row reinforcement is recommended. For the shear wall with a thickness exceeding 700 mm, quadruple-row reinforcement is advised.

$$\begin{cases} n = 2, & t \le 400 \\ n = 3, & 400 < t \le 700 \\ n = 4, & t > 700 \end{cases} \tag{14}$$

where n is the number of reinforcement rows.

The reinforcement ratio of longitudinal and transverse bars in shear walls should not be less than 0.25 % in seismic design (i.e., grades I, II and III) and 0.20 % in non-seismic design (i.e., grade IV).

$$\begin{cases} \rho_{\min} \le \rho_v \\ \rho_{\min} \le \rho_h \end{cases} \tag{15}$$

Since the agent optimizes the design toward minimum material cost, excessively small spacing, which would substantially increase reinforcement usage, is automatically avoided through the cost function. Consequently, the minimum spacing constraint is retained in Eq. (16) to ensure structural safety. According to Chinese design code [41,42] the spacing of longitudinal and transverse bars in shear walls should not exceed 300 mm, and the rebar diameter should not be less than 6 mm, but it should not exceed 1/10 of the wall thickness.

$$\begin{cases} S_{\min} \le S_v \\ S_{\min} \le S_h \end{cases} \tag{16}$$

3.3.4. Stability

According to the design codes [41,42], the stability of shear walls should meet the following conditions:

$$q \le \frac{E_c t^3}{10 l_0^2} \tag{17}$$

where q is the axial force acting on the top of the wall; E_c is the elastic modulus of the shear wall concrete; l_0 is the effective length ($l_0 = \beta h$); β is the effective length factor of the wall section (= 1.0 in this study); and h is the story height.

3.3.5. Axial compression ratio

The axial compression ratio of shear walls should satisfy Eq. (18). For seismic grades I and II, the ratio (μ) is set at 0.5, while for grade III, it is 0.6.

$$\mu \ge N/(A \cdot f_c) \tag{18}$$

where N is the axial load; and A is the cross-sectional area of the shear wall.

3.3.6. Economic consideration

According to Goldberg [52], for cost minimization problems, the objective cost should be subtracted from a large fixed value, so that all fitness values become positive. The fitness calculation method is defined as follows:

$$C_{ex} = C_{\cos t} - C_m \tag{19}$$

where C_{ex} is the excess amount after expenses. Minimizing C_{ex} corresponds to minimizing the actual cost. $C_{\cos t}$ is the actual expense, which is calculated from the cost of the concrete and reinforcement (i.e., determined by their respective volumes multiplied by unit prices); and C_m is the minimum cost of shear wall under a given shear force.

To investigate the possible material cost associated with the shear strength of shear walls under different sizes and loads, all possible combinations of wall sizes and loads were generated to calculate the shear strength and costs. Table 3 shows the design parameter range that satisfies the design code requirements. The used price is 59.86 \$/m^3 for concrete and 17500 \$/m^3 for steel reinforcement. The final linear relationship between the shear strength and materials cost is defined as follows (Fig. 6):

$$C_m = 0.02641 V_n + 18.12 \tag{20}$$

Although the enumeration method can provide a useful reference for the cost-shear strength relationship, the current case studies rely on sparsely sampled design points and are computationally time-consuming; moreover, for high-dimensional structural systems, a lookup-table-based enumeration approach becomes impractical for real

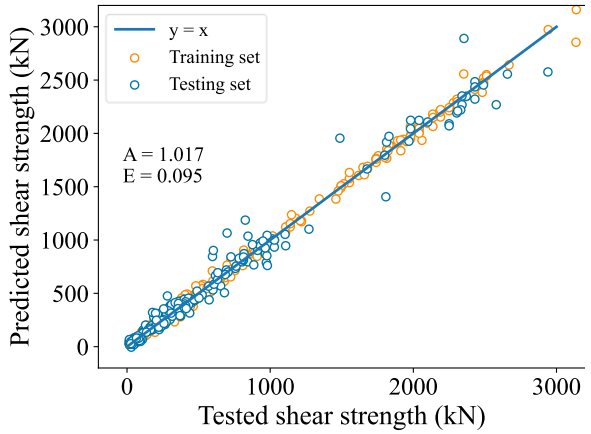


Fig. 5. Accuracy of the XGBoost-based learning framework for predicting V_u.

Table 3
Parameter ranges to estimate C_m.

Parameters	d_b (mm)	N	b_w (mm)	t (mm)
Range	6–28	4–80	500–2100	100–300

Notes: d_b is the rebar diameter; N is the number of transverse bars; and b_w and t are the width and thickness of shear wall, respectively.

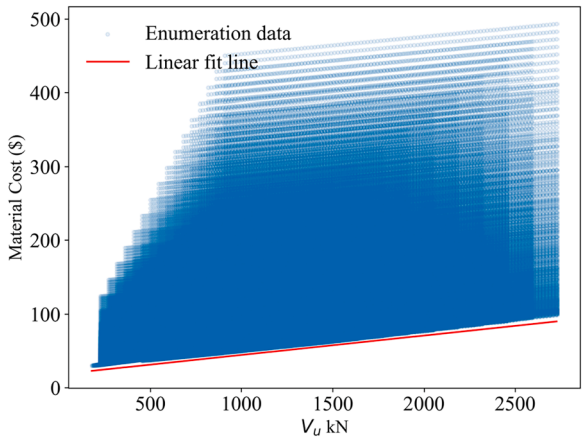


Fig. 6. Material cost in accordance with shear strength.

engineering applications.

3.4. Reward

For optimizing the model during the design process, it is crucial to define an appropriate reward function. The reward function can be regarded as an objective function for the DRL agent, mapping the features of the shear wall to the reward. For this reason, the reward function contains the prior knowledge of the design task's objectives, including compliance with the design code requirements and cost-effectiveness. Each requirement is evaluated by a separate reward function, and the total reward is estimated from the weighted sum of the individual rewards.

A straightforward approach is adopted using linear functions with varying gradients to generate the reward. A value of 0.0 serves as the baseline, indicating that the design code requirements have been allowed. If a feature does not meet the code requirement, a penalty (i.e., negative reward) is added based on the distance to the requirement boundary. The calculation steps of the reward can be seen in Fig. 7. The reward function to ensure compliance with the design code requirements can be defined as follows:

$$R = \begin{cases} \text{negativereward} & \text{ifparametersnotmeetcoderequirements} \\ \text{positivereward} & \text{materialscost} \end{cases} \tag{21}$$

3.4.1. Rewards for design code requirements (negative reward)

DRL aims to maximize rewards, and the structure should first satisfy the design code requirements when shear wall structures are designed. Thus, if the structural parameters do not satisfy the design code or cost requirements, a corresponding penalty value should be set (i.e., the reward will be negative). Also, the current learning episode will not be terminated until the maximum trial action steps reach. The penalty can be calculated from Eq. (22). The reward is affected by the severity of the violations, with larger penalties for more significant deviations from the requirements.

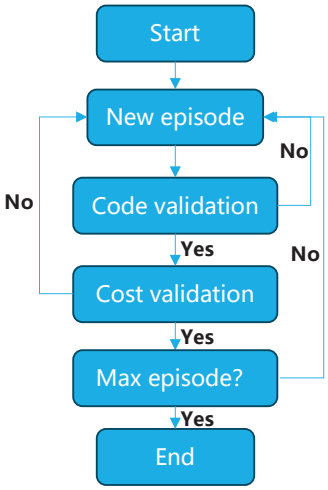


Fig. 7. Reward calculation steps.

$$R_{\text{std}} = \sum_{i=1}^{n} \omega_i \cdot R_i \qquad \text{for } R_i < 0 \tag{22}$$

where R_{std} is the weighted sum of all negative reward values; ω_i is the weighting coefficient for each negative reward (after multiple attempts, $\omega_1 \sim \omega_6$ are set to 0.2, 0.2, 0.2, 0.1, 0.3, and 3.0, respectively, in this study); and R_i refers to the evaluation value of design code requirements, which corresponds to the wall dimensions, rebar spacing, rebar ratio, stability, axial compression ratio, and shear strength. The negative reward range is [-∞ , 0]. The specific definitions of R_i are defined as follows:

For wall dimensions, the reward value R_1 is as follows:

$$R_1 = \left(\frac{t - t_{\min}}{t_{\min}}\right) \tag{23}$$

where t is the thickness of the shear wall.

For rebar spacing, the reward value R_2 is calculated from the number of rebar layers and the width and thickness of the shear wall.

$$R_2 = \left(\frac{S_{\max} - S}{S_{\max}}\right) \tag{24}$$

where $S_{\max}$ is 300 mm [42]; and S is the maximum rebar spacing for the longitudinal reinforcement and stirrups, which can be calculated based on w, c, n, and N_t. b_w is the shear wall width; c is the concrete cover and is assumed to be 30 mm; n is the number of rebar layers in the shear wall; and N_t indicates the total number of rebars in the corresponding direction.

For rebar ratio, the reward value R_3 is calculated from the longitudinal and transverse rebar ratios.

$$R_3 = \left(\frac{\rho - \rho_{min}}{\rho_{min}}\right) \tag{25a}$$

$$\rho = \min(A_{s,v}/wt\ ,\ A_{s,h}/ht) \tag{26b}$$

where $A_{s,v}$ and $A_{s,h}$ are the total cross-sectional area of longitudinal and transverse rebars, respectively.

For stability, the reward value R_4 is as follows:

$$R_4 = \frac{\frac{E_c t^3}{10 l_0^2} - q}{q} \tag{27}$$

where q is the axial force of the wall (Eq. (17)).

For axial compression ratio μ, the reward value R_5 is as follows:

$$R_5 = \left[\frac{\mu - N/(A \cdot f_c)}{\mu}\right] \tag{28}$$

For shear demand V_u, the reward value R_6 is as follows:

$$R_6 = \left[\frac{V_n - V_u}{V_u}\right] \tag{29}$$

where V_n is nominal shear strength calculated by design code or XGBoost.

3.4.2. Material cost rewards (positive reward)

After design code requirements are evaluated, the reward for the

material cost also needs to be estimated. The primary goal of this study is to control the design cost of shear walls. Ensuring that the DRL framework retains sufficient exploration capability while maintaining economic efficiency. Therefore, adopting a cost limit of 1.2 times [33] the minimum cost (C_m) represents a slightly more conservative yet practical assumption. In this case, when the structural design satisfies the code requirements, an economic efficiency check will be performed, otherwise a negative reward is given. The reward value R_{cost} is defined as follows:

$$R_{\cos t} = 100 \cdot \frac{0.2}{\frac{C_{ex}}{C_m} + 0.2} \tag{30}$$

where R_{cost} is a value relative to C_m and C_{ex}; and C_m is the minimum material cost under a given shear force, which can be calculated from Eq. (20). If the material cost (C_{cost}) exceeds $1.2C_m$, the reward range is restricted to [0, 50]. When the material cost is less $1.2C_m$, the reward will be positive, showing the range of (50,100].

3.5. Training and test setup

For training, the parameters and combined hyperparameters were empirically set (Table 4). Two fully connected layers were adopted with 128 and 128 neurons for both actors and critics. The critic's output layer has a linear activation function, while the actor's output layer has a tanh activation function. During each episode, a random shear force ranging from 200 to 1500 kN is generated, and up to 10 actions are executed in sequence until a reasonable structural design is achieved. The actor and critic networks were trained during 100,000 episodes. It is important to note that during the training process, the DRL agent will generate a large amount of data through continuous exploration, which is stored in a replay buffer. A portion of this data is subsequently used for policy learning. Thus, it is necessary to restrict the generation of uninformative or unproductive data. After several trials, a method was adopted in which any design resulting in a shear wall cost greater than twice the minimum observed cost is considered invalid, and the current episode is terminated. This approach ensures that the agent retains sufficient space for exploration while also guaranteeing that the collected data remains meaningful for accelerating policy learning and avoiding convergence to suboptimal local solutions. All simulations were performed on a computer equipped with an Intel Core i7–12700 CPU, 16 GB RAM, and Windows 11 operating system. The algorithms were implemented in Python 3.10 using the PyTorch deep learning framework.

4. Results and discussion

4.1. Training results

Fig. 8 shows the historical rewards after 100,000 episodes of training, in which each point represents the total reward obtained by SAC in an episode. For training of the both methods, the total training time was approximately 80 min. It is evident that around episode 50,000, the algorithm using code requirements (i.e., Method 1) begins to be converged. However, using the XGBoost-method (i.e., Method 2), convergence was achieved after approximately 40,000 iterations. When Method 1 was adopted, within the first 50,000 steps, the algorithm was still in the early stages of learning, during which the policy remained immature and the balance between exploration and exploitation had not yet been established. After this period, the algorithm began to discover relatively stable policies. The average reward for the shear wall reached about 50, which means the designed requirements are satisfied and the material cost (C_{cost}) is close to $1.2C_m$. At this point, the shear wall design also satisfied the design code requirements. The reward progressively increased with the number of episodes, indicating continuous policy improvement. When using the XGBoost-based method, the policy initially performed poorly during the first 20,000 iterations, resulting in low rewards. In the subsequent stages, the total reward tended to an upward trend and gradually stabilized, suggesting that the algorithm had converged and identified a relatively optimal policy. Compared with Method 1, Method 2 exhibits lower volatility and better stability, achieving a higher final reward level with a more favorable upward trend. The use of the XGBoost-based method demonstrates clear advantages. This superiority may stem from the fact that the code-based formulation in Method 1 lacks sufficient sensitivity to parameter variations. Consequently, any changes in parameters would necessitate rule redesign or model adjustment, thereby limiting the adaptability and robustness of the DRL model. This observation demonstrates SAC's ability to design shear wall structures that meets the strength requirements as well as optimizes cost, achieving a balance between the safety and economic efficiency.

Table 4
Parametric setting in the simulation.

Item	Description
Action dim	6
State dim	9
Learning rate of actor	1e-4
Learning rate of critic	1e-3
Gamma	0.95
Tau	0.005
Buffer size	30000
Batch size	128

After training, the SAC agent was tested on 110 randomly generated design cases. The DRL model produced 100 shear wall structures subjected to randomly generated V_u ranging from 200 to 1500 kN, and an additional 10 designs with V_u between 1500 and 1700 kN. The trained SAC successfully designed all cases without any violations, and saved material costs. For each design case, Fig. 9 shows the reward values for all successful cases. It can be observed that the design cost increases as the shear demand of the shear wall increases. The variation trends of shear wall parameters designed using both methods were generally consistent with respect to the shear demand, except for the amount of longitudinal reinforcement. This discrepancy is because the design code formulas do not consider the contribution of longitudinal reinforcement to the shear strength. On the other hand, the XGBoost training results indicated that longitudinal reinforcement affected the shear wall design. Moreover, since the accuracy of XGBoost predictions heavily depends on the quantity and diversity of the training data, the trend between the number of transverse bars and the shear demand V_u (Fig. 10**(e)**) does not exhibit a clear linear pattern in some automated designs. This issue was primarily attributed to insufficient training data or the need for additional training iterations. As shown in Figs. 9**(a)** and 10**(a)**, for both methods, the shear strengths of the designed walls were consistently greater than the shear demands, generally maintaining a margin within 50 %. This indicates that no excessive shear strength design occurs, which contributes to reduced material consumption and improved structural efficiency. The shear wall width (b_w) and thickness (t) also increased with the increase in shear demand. According to the strength calculation formulas (Eqs. (10) and (11)), the most effective way to improve the shear strength is to enlarge the cross-sectional area. Additionally, in order to satisfy the maximum spacing requirements, the amount of longitudinal reinforcement increased accordingly as the wall width increased. In terms of reinforcement configuration, the reinforcement ratio is mainly increased by enlarging the amount of rebar, while the diameter of rebars is constant at 6 mm. From Figs. 9**(f)** and 10**(f)**, it can be seen that the design rewards are all above 50, indicating that the designs meet the requirements while keeping the cost within 1.2 C_m. Further validation on shear demands range from 1500 to 1700 kN showed that the DRL agent sustained stable and reliable performance, demonstrating its ability to generalize beyond the training range. As shown in Fig. 11, the policy can be visualized and interpreted by human design engineers. States and actions form a path through the solution

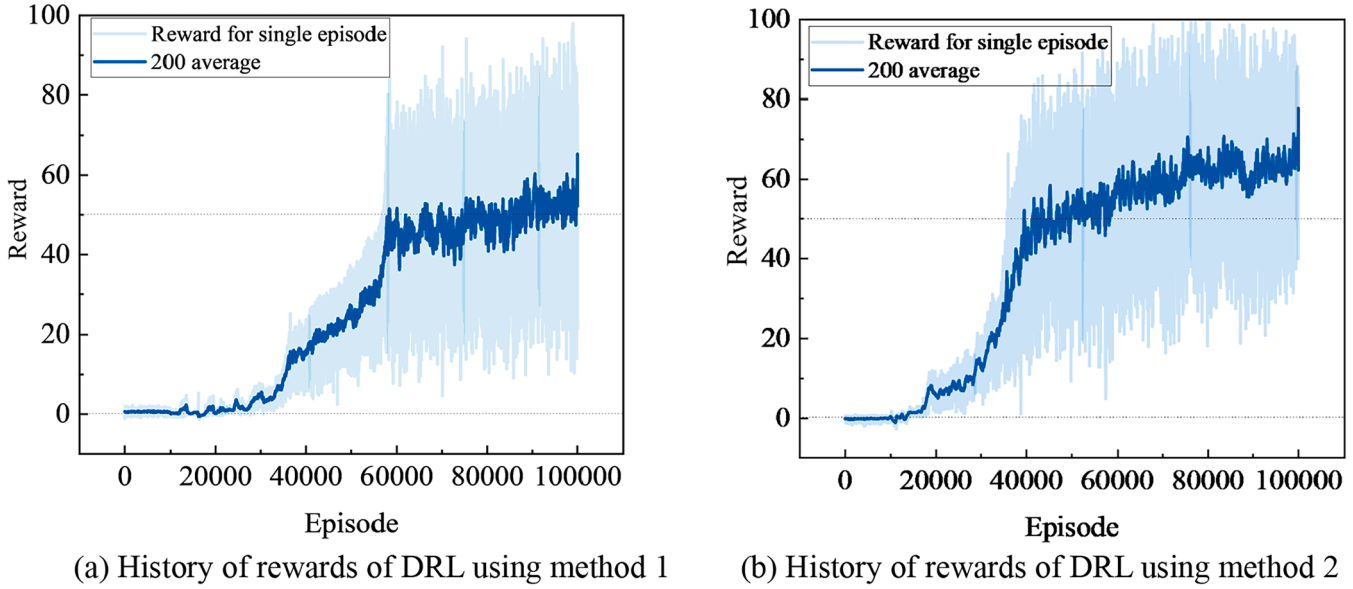


Fig. 8. Reward variations according to episodes.

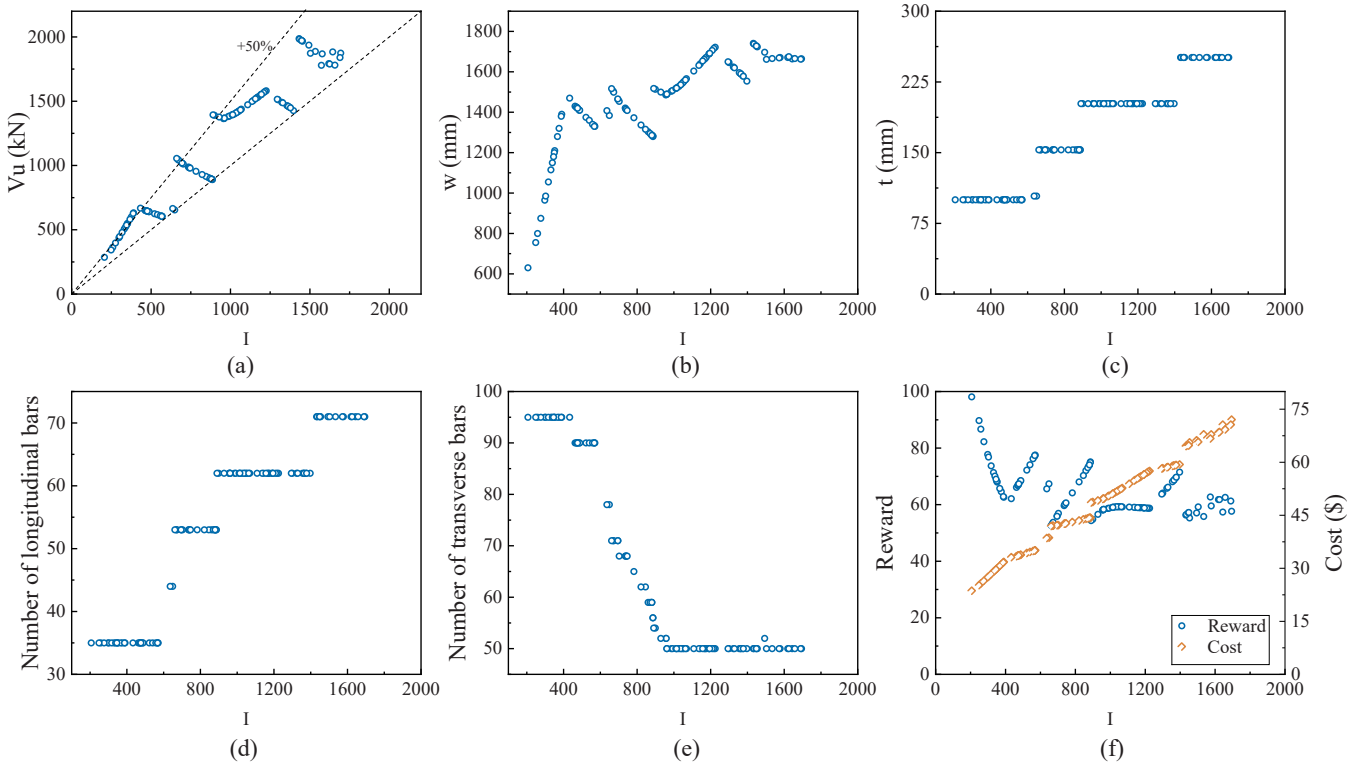


Fig. 9. Test results of 110 random shear walls designed using SAC-M1.

space which provides insights of the learned behavior. Since the thickness (t) in this study was determined based on the data range used for training the XGBoost-based method, some of the generated design cases may not fully comply with practical engineering requirements. Nevertheless, certain results still possess valuable engineering significance.

4.2. Design comparison with genetic algorithm

Atabay [12] used a genetic algorithm (GA) to optimize the cost of RC shear walls, treating shear wall dimensions as design variables. The objective was to find the optimal shear wall dimensions that minimize total material cost, and a corresponding computer program was developed for optimization calculations. Method 1 and Method 2 produced similar results in automated design, so that only Method 2 was selected for comparison with the GA algorithm. The Genetic Algorithm (GA) was configured with a population size of 200 and 50 generations. The crossover and mutation probabilities were set to 0.7 and 0.2, respectively, using a blend crossover operator (crossover weight=0.5). The fitness function was defined consistently with the reward function of the DRL model. Fig. 12 shows the shear wall system that needs to be

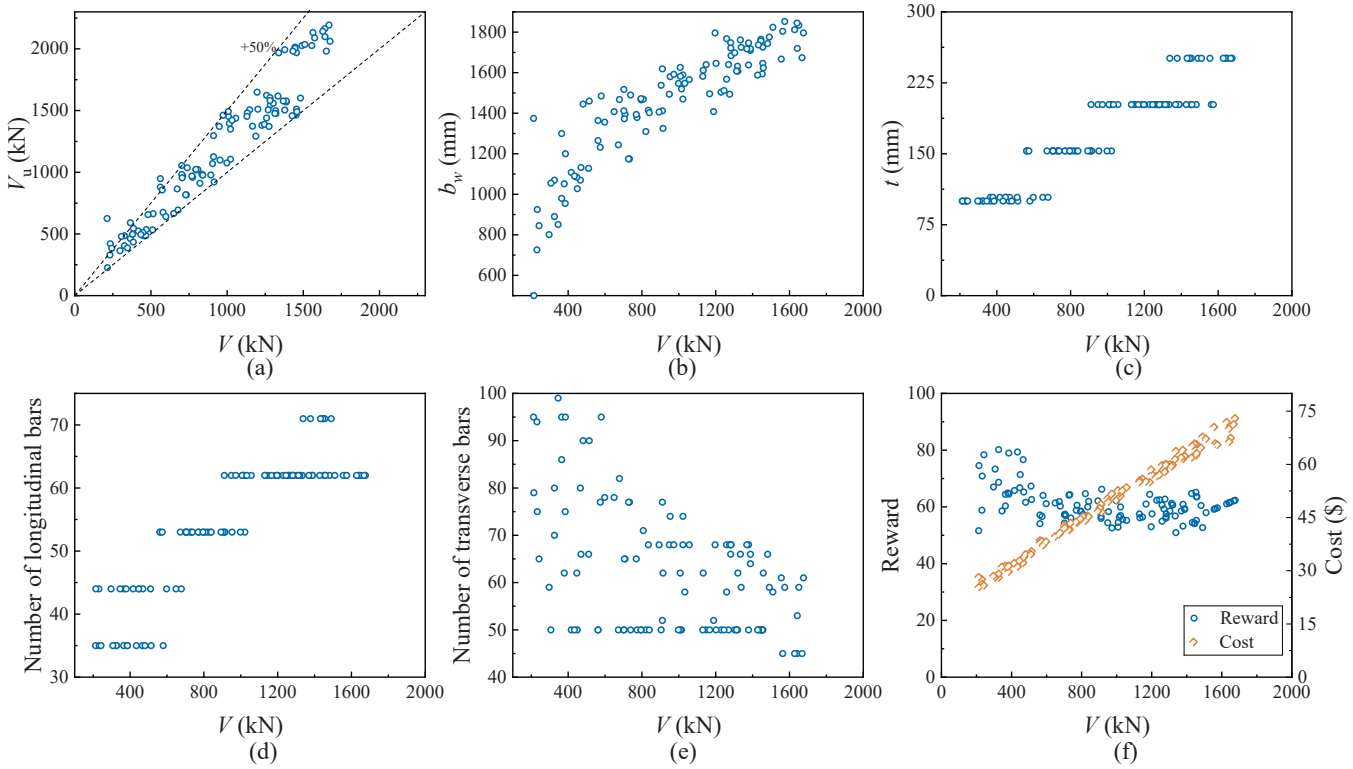


Fig. 10. Test results of 110 random shear walls designed using SAC-M2.

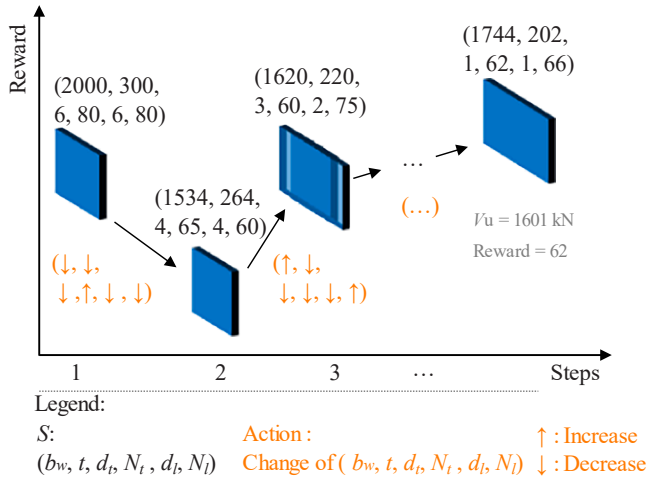


Fig. 11. Illustration of a trained policy as path through the solution space formed by states and actions. (notes: d_l and d_t are the diameter of longitudinal and transverse bars, respectively; N_l and N_t are the number of longitudinal and transverse bars, respectively.).

optimized. The floor has no beams, and each floor has a total of 10 shear walls. The height of the shear wall is assumed to be constant, and the shear demand and axial force acting on the shear wall are known. Both DRL and GA aim to design shear walls by finding the optimal values for the shear wall width, thickness, and the index and number of transverse reinforcements. This study used Atabay [12] method and layout of shear wall in plan, but the material cost and shear wall constraints were determined from the methods in Section 3.

Table 5 and Table 6 compare the shear wall structures generated by the DRL and GA, respectively, both of which meet the design code requirements. And the variation of the design parameters with V_u is shown in Fig. 13. From the shear walls generated by both the DRL and GA, it is evident that the shear walls designed by DRL are more economical, with SW9 exceeding the budget of the GA-designed shear walls, the differences remain minimal. In terms of shear wall design, both methods share the objective of minimizing the cross-sectional dimensions and the amount of reinforcement used. Both methods identified that minimizing the rebar diameter could effectively reduce the overall cost. In the case of small cross-sections, both approaches also demonstrated an increase in the cross-sectional area with the growth of shear force V_u. For example, in the design of SW2 and SW4, which have similar shear demands, the parameters generated by the DRL-based design approach are nearly identical. In contrast, the shear wall parameters optimized by the GA algorithm exhibit significant variation. A similar phenomenon can also be observed in SW9 and SW10. This discrepancy is primarily attributed to the inherent randomness of the GA optimization process. DRL was able to learn design strategies from historical training data. In contrast, the GA method performed noticeably worse than DRL in generating appropriate wall thickness and the numbers of transverse bars. This is primarily because the GA often becomes trapped in local optima during the optimization process, which is a common issue when solving high-dimensional optimization problems. It is worth noting that the performance of GA is also heavily influenced by the choice of encoding methods and hyperparameter settings. During the process of automatically generating 10 shear walls, DRL took 0.1 s, while the GA took about 200 s, showing a significant improvement in computational efficiency with DRL compared to the GA.

4.3. Extension in I-shape shear wall

To demonstrate the potential of DRL in the design of more complex shear wall configurations, such as I-shaped walls, the DRL framework was retrained for the automated design of non-rectangular shear walls. The cross-section of the I-shaped wall is shown in Fig. 14 (a). The design variables for the wall geometry include the wall web thickness (S1), wall width (S2), flange width (S3), and flange thickness (S4). The reinforcement-related design variables include the longitudinal and

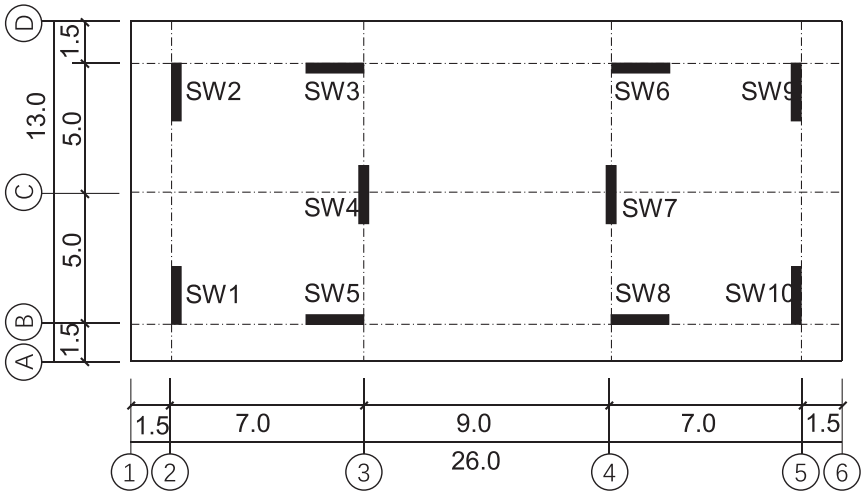


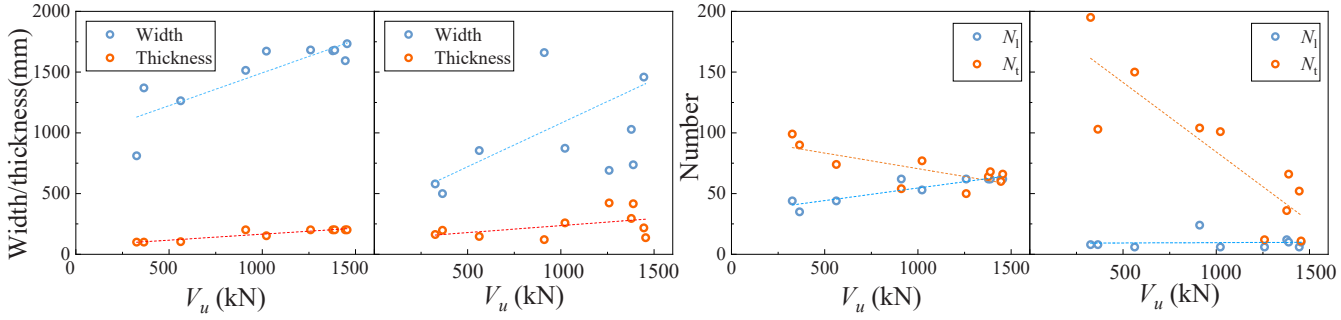
Fig. 12. Layout of shear wall in plan (unit: m).

Table 5
Structural parameters of the shear wall generated by Method 2.

Item	SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8	SW9	SW10
Width (mm)	1682	1250	1370	1380	1673	811	1264	1315	1603	1734
Thickness (mm)	202	240	100	200	153	100	104	202	253	202
d_l (mm)	6	6	6	6	6	6	6	6	6	6
N_l	52	55	35	45	53	44	44	62	32	38
d_t (mm)	6	6	6	6	6	6	6	6	6	6
N_t	50	55	90	57	77	99	74	54	42	36
V_n (kN)	1545	1396	623	1443	1164	369	598	1390	1464	1592
V_u (kN)	1259	1378	365	1388	1022	326	562	912	1445	1455
Material cost ($)	57.7	46.8	39.7	49.5	54.6	34.1	36.6	47.3	49.3	53.3

Table 6
Structural parameters of the shear wall generated by GA.

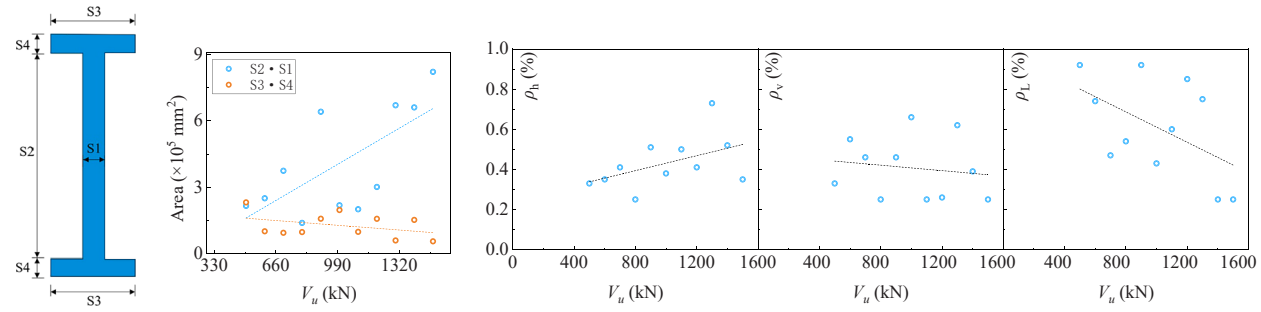
Item	SW1	SW2	SW3	SW4	SW5	SW6	SW7	SW8	SW9	SW10
Width (mm)	1089	1029	500	737	873	691	854	1661	1459	2488
Thickness (mm)	208	295	198	416	259	275	147	121	218	137
d_l (mm)	6	6	6	6	6	6	6	6	6	6
N_l	52	12	8	10	6	20	6	24	8	10
d_t (mm)	8	8	20	10	8	8	12	6	6	16
N_t	63	36	103	66	101	47	150	104	52	11
V_n (kN)	1295	1380	450	1394	1028	370	571	914	1446	1550
V_u (kN)	1259	1378	365	1388	1022	326	562	912	1445	1455
Material cost ($)	58.1	47.8	74.5	50.5	63.5	35.5	73.5	47.5	44.5	62.3



(a) V_u versus wall width and thickness generated by DRL (left) and GA (right)

(b) V_u versus N_l and N_t generated by DRL(left) and GA(right)

Fig. 13. Parameters vs Vu generated by DRL and GA.



(a) Cross section of I-shape shear wall | (a) Web and flange areas | (b) Transverse reinforcement ratios | (c) Longitude reinforcement ratios | (d) Flange reinforcement ratios

Fig. 14. Designed cross section parameters of I-shape shear wall by DRL.

transverse reinforcement ratios of the web (ρ_v and ρ_h), as well as the reinforcement ratio of the flange (ρ_L). In total, the DRL model optimizes seven design parameters.

The target shear capacity for the I-shaped walls was set within the range of 500–1500 kN, DRL **Method 2** was employed for training. A total of 100,000 episodes were executed. Upon completion of the training, two I-shaped shear walls were randomly designed for validation. The results are presented in Fig. 14. The generated dimensions S1–S4 show no clear correlation with V_u. However, the web area (S1 ×S2) increases with higher V_u, while the flange area (S3 ×S4) decreases as V_u increases. This trend is consistent with Eq. (11), where under high shear force(V_u) the value of A_w/A increases, thereby enhancing the contribution of axial force to the overall shear capacity. As the V_u increases, the horizontal reinforcement ratio shows a noticeable growth, whereas the vertical reinforcement ratio remains nearly constant. The flange reinforcement ratio also exhibits a general decreasing tendency. The results indicate that the DRL framework originally developed for rectangular shear wall design can be readily extended to I-shaped configurations, achieving short design times while fully satisfying code requirements. Overall, the DRL-based shear wall design approach demonstrates strong scalability, being capable of designing both rectangular and non-rectangular shear walls, although the limited availability of I-shaped wall data in the database still imposes certain constraints on the accuracy of XGBoost-based predictions for such configurations.

5. Discussion

DRL is capable of automating design tasks by finding policies that lead from initial to suitable parametric designs. However, several challenges arise when employing deep reinforcement learning (DRL) for the design of building structures. As previously mentioned, DRL involves a multitude of hyperparameters, which need to be re-optimized for different training tasks, thereby increasing the workload associated with DRL. Moreover, even with identical hyperparameter settings, the final models would still vary in different training tasks due to the stochastic nature of DRL's exploration in the solution space. Additionally, in this study, the structures designed using DRL are not yet fully mature for practical engineering applications and require further constraints and training to ensure that the shear walls meet real-world requirements. For example, the designed walls may have relatively large widths or reinforcement spacings that differ from typical engineering practice. Therefore, incorporating the experience of skilled structural engineers is essential to introduce more practical constraints during training and achieve more realistic and applicable design solutions. Lastly, a single training task often necessitates a large number of episodes, each comprising numerous steps, which results in substantial computational resource consumption. This makes the computational cost of individual calculations or model optimizations quite high.

Nevertheless, DRL also demonstrates considerable potential widespread application in the design of building structures. Most notably, for design tasks involving a large number of repetitive structures. DRL can rapidly generate design outcomes after a single training process, thereby significantly enhancing engineering design efficiency. Similarly, the proposed framework can be extended to automated design under other design codes by adjusting the detailed provisions specified in the respective standards. Additionally, achieving a fully automated design would require incorporating more action variables into the training process, which would, however, require significantly more computational resources. Moreover, pre-training the DRL model [53,54] using existing design data can improve the accuracy and rationality of the design results. In addition to enabling the automated design of structures, DRL also excels in the optimization of existing schemes. The agent evolves structural parameters from an initial state through dynamic interaction with the environment, ultimately yielding designs that satisfy all constraints. Once trained, the DRL model can be directly applied to diverse working conditions or initial schemes, drastically reducing repetitive iterations and computational costs, and thus outperforming traditional meta-heuristic algorithms in optimization efficiency.

6. Conclusion

In the present study, deep reinforcement learning (DRL) using design code and XGBoost was applied to propose an automated design method for shear wall. The approach adopts a conventional DRL framework, consisting of an agent, environment, and their interactions through actions, rewards, and states. The automatic design method for shear walls employs a trial-and-error approach to autonomous learning, interacting with defined normative rewards and load resistance rewards, aiming to maximize the reward and generate reasonable structures. After 100,000 training episodes, the DRL successfully designed shear walls, and it was further validated through testing on 110 shear wall structures. Compared with traditional optimization algorithms, the computational results demonstrate the distinct advantages of the DRL, including faster computation and significant cost savings. The primary conclusions are as follows:

(1) The SAC algorithm is used to design shear wall structures, and a reward function and computational framework for shear wall design are established. The method provides reasonable design and economic optimization of shear wall structures while ensuring compliance with design code.
(2) For training DRL for shear wall design, the design code-based method and XGBoost-based method were applied. Both methods enable fast and accurate structural design, offering valuable physical knowledge and experience to automated design of shear wall structures. The XGBoost-based method achieves

convergence within fewer episodes, but the method depends on the quantity and quality of the collected data. In contrast, the design code-based method without data collection exhibits slow convergence, but reveals relatively reasonable design results.

(3) Both the DRL and GA methods were used to design 10 shear walls. Compared with traditional GA method, the DRL method designs demonstrated greater rationality and applicability to real world engineering practice. Specifically, the DRL-based shear wall design method was able to complete the design process within 0.1 s, leveraging the design experience and strategies acquired during the training phase. The DRL method is more suitable for design tasks that involve a large number of repetitive operations.

(4) The DRL-based automated design of shear walls can be extended from rectangular configurations to I-shaped shear walls, demonstrating the transferability of the DRL approach.

The present study primarily focuses on the automated design of simple shear wall structures, serving as an initial exploration. The structural forms considered are relatively simple, and only material cost was taken into account. In practical applications, however, additional constraints should be considered. Moreover, the development of automated design methods for integrated structures composed of shear walls, beams, and columns still requires further investigation.

CRediT authorship contribution statement

Hyeon-Jong Hwang: Writing – review & editing, Supervision, Investigation, Funding acquisition. **Gao Ma:** Project administration, Funding acquisition, Formal analysis. **Min-Jeong Cho:** Supervision, Investigation. **Guoxu Zhao:** Writing – original draft, Visualization, Software, Methodology, Data curation, Conceptualization.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationship that could have appeared to influence the work reported in this paper.

Acknowledgment

The present research was supported by the National Natural Science Foundation of China (Grant No. 52278498 and No. 51878268), the Huxiang Youth Talent Support Program of Hunan Province, (Grant No. 2021RC3041), the Natural Science Foundation of Hunan Province, China (Grant No. 2020JJ4195), and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. RS-2024–00333882).

Data availability

Data will be made available on request.

References

[1] Berquand A., Murdaca F., Riccardi A., Soares T., Geren S., Brauer N. et al. Towards an artificial intelligence based design engineering assistant for the early design of space missions. 69th International Astronautical Congress. DEU. p. 554574.
[2] De Boissieu A. Introduction to computational design: subsets, challenges in practice and emerging roles. In: Bolpagni M, Gavina R, Ribeiro D, editors. Industry 40 for the Built Environment: Methodologies, Technologies and Skills. Cham: Springer International Publishing; 2022. p. 55–75.
[3] Hammad A, Itoh Y, Nishido T. Bridge planning using gis and expert system approach. J Comput Civ Eng 1993;7:278–95.
[4] Rafiq MY, Bugmann G, Easterbrook DJ. Neural network design for engineering applications. Comput Struct 2001;79:1541–52.
[5] Ballal TMA, Sher WD. Artificial neural network for the selection of buildable structural systems. Eng Constr Archit Manag 2003;10:263–71.
[6] Jootoo A, Lattanzi D. Bridge type classification: supervised learning on a modified NBI data set. J Comput Civ Eng 2017;31:04017063.
[7] Sigmund O, Maute K. Topology optimization approaches. Struct Multidiscip Optim 2013;48:1031–55.
[8] Lei X, Liu C, Du Z, Zhang W, Guo X. Machine learning-driven real-time topology optimization under moving morphable component-based framework. J Appl Mech 2018;86.
[9] Yu Y, Hur T, Jung J, Jang IG. Deep learning for determining a near-optimal topological design without any iteration. Struct Multidiscip Optim 2019;59:787–99.
[10] Zhang Y, Mueller C. Shear wall layout optimization for conceptual design of tall buildings. Eng Struct 2017;140:225–40.
[11] Lou H, Xiao Z, Wan Y, Quan G, Jin F, Gao B, et al. Size optimization design of members for shear wall high-rise buildings. J Build Eng 2022;61:105292.
[12] Atabay Ş. Cost optimization of three-dimensional beamless reinforced concrete shear-wall systems via genetic algorithm. Expert Syst Appl 2009;36:3555–61.
[13] Cerè G, Rezgui Y, Zhao W, Petri I. Shear walls optimization in a reinforced concrete framed building for seismic risk reduction. J Build Eng 2022;54:104620.
[14] Lou HP, Ye J, Jin FL, Gao BQ, Wan YY, Quan G. A practical shear wall layout optimization framework for the design of high-rise buildings. Structures 2021;34:3172–95.
[15] 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.
[16] Alanani M, Brown T, Elshaer A. Automated shear wall layout optimization framework of tall buildings subjected to dynamic wind loads. In: Desjardins S, Poitras GJ, El Damatty A, Elshaer A, editors. Proceedings of the Canadian Society for Civil Engineering Annual Conference 2023, Volume 13. Cham: Springer Nature Switzerland; 2024. p. 285–99.
[17] Qin S-Z, Guan H, Liao W-J, Gu Y, Zheng Z, Xue H-J, et al. An artificial intelligent design system for shear wall structures with large language model controlling generation and optimization. In: Francis A, Miresco E, Melhado S, editors. Advances in Information Technology in Civil and Building Engineering. Cham: Springer Nature Switzerland; 2025. p. 227–37.
[18] Qin S, Liao W, Gu Y, Zheng Z, Xue H, Lu X. Intelligent design and optimization system for shear wall structures based on large language models and generative artificial intelligence. J Build Eng 2024;95:109996.
[19] Wang Z, Yu Y, Chen Y, Liao W, Li C, Hu K, et al. Expert experience-embedded evaluation and decision-making method for intelligent design of shear wall structures. J Comput Civ Eng 2024;39.
[20] Chen J, Bao Y. Multi-agent large language model framework for code-compliant automated design of reinforced concrete structures. Autom Constr 2025;177:106331.
[21] Qin S, Guan H, Liao W, Gu Y, Zheng Z, Xue H, 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.
[22] Wang W, Zmeureanu R, Rivard H. Applying multi-objective genetic algorithms in green building design optimization. Build Environ 2005;40:1512–25.
[23] Lutz ID, Wang S, Norn C, Courbet A, Borst AJ, Zhao YT, et al. Top-down design of protein architectures with reinforcement learning. Science 2023;380:266–73.
[24] Strokach A, Becerra D, Corbi-Verge C, Perez-Riba A, Kim PM. Fast and flexible protein design using deep graph neural networks. Cell Syst 2020;11:402–11. .e4.
[25] Dworschak F, Dietze S, Wittmann M, Schleich B, Wartzack S. Reinforcement learning for engineering design automation. Adv Eng Inform 2022;52:101612.
[26] Mnih V, Kavukcuoglu K, Silver D, Rusu AA, Veness J, Bellemare MG, et al. Human-level control through deep reinforcement learning. Nature 2015;518:529–33.
[27] Lee S, Bang H. Automatic gain tuning method of a Quad-Rotor geometric attitude controller using A3C. Int J Aeronaut Space Sci 2020;21:469–78.
[28] Kiran BR, Sobh I, Talpaert V, Mannion P, Sallab AAA, Yogamani S, et al. Deep reinforcement learning for autonomous driving: a survey. IEEE Trans Intell Transp Syst 2022;23:4909–26.
[29] Pan Y, Shen Y, Qin J, Zhang L. Deep reinforcement learning for multi-objective optimization in BIM-based green building design. Autom Constr 2024;166:105598.
[30] Minsky M. Steps toward artificial intelligence. Proc IRE 1961;49:8–30.
[31] Brown NK, Garland AP, Fadel GM, Li G. Deep reinforcement learning for engineering design through topology optimization of elementally discretized design domains. Mater Des 2022;218:110672.
[32] Cheng G, Zhou X, Liu J, Wang L. Intelligent design method of high-rise shear wall structures based on deep reinforcement learning. J Build Struct 2022;43:84–91.
[33] Jeong J-H, Jo H. Deep reinforcement learning for automated design of reinforced concrete structures. ComputAided Civ Infrastruct Eng 2021;36:1508–29.
[34] Hayashi K, Ohsaki M. Reinforcement learning for optimum design of a plane frame under static loads. Eng Comput 2021;37:1999–2011.
[35] Hayashi K, Ohsaki M. Graph-based reinforcement learning for discrete cross-section optimization of planar steel frames. Adv Eng Inform 2022;51:101512.
[36] Fu B, Gao Y, Wang W. A physics-informed deep reinforcement learning framework for autonomous steel frame structure design. ComputAided Civ Infrastruct Eng 2024;39:3125–44.
[37] Moghaddas SA, Bao Y. Explainable machine learning framework for predicting concrete abrasion depth. Case Stud Constr Mater 2025;22:e04686.
[38] Ma G, Qin C, Hwang H-J, Zhou Z. Data-driven models for predicting tensile load capacity and failure mode of grouted splice sleeve connection. Eng Struct 2023;289:116236.
[39] Haarnoja T, Zhou A, Abbeel P, Levine S. Soft Actor-Critic: off-policy maximum entropy deep reinforcement learning with a stochastic actor. In: Jennifer D, Andreas K, editors. Proceedings of the 35th International Conference on Machine Learning. Proceedings of Machine Learning Research. PMLR; 2018. p. 1861–70.

[40] Raziei Z, Moghaddam M. Adaptable automation with modular deep reinforcement learning and policy transfer. Eng Appl Artif Intell 2021;103:104296.
[41] Code for seismic design of buildings. China: China Architecture & Building Press. 2010.
[42] Technical specification for concrete structures of tall building. Chinese 2010.
[43] Committee ACI. Building code requirements for structural concrete: (ACI 318-25). Farmington Hills, MI: American Concrete Institute; 2025. ©2025; 2025.
[44] Hwang S-J, Fang W-H, Lee H-J, Yu H-W. Analytical Model for Predicting Shear Strengthof Squat Walls. J Struct Eng 2001;127:43–50.
[45] Chen T. XGBoost: A Scalable Tree Boosting System. Cornell University; 2016.
[46] Zhao X, Lovreglio R, Nilsson D. Modelling and interpreting pre-evacuation decision-making using machine learning. Autom Constr 2020;113:103140.
[47] Feng D-C, Wang W-J, Mangalathu S, Taciroglu E. Interpretable XGBoost-SHAP Machine-Learning Model for Shear Strength Prediction of Squat RC Walls. J Struct Eng 2021;147:04021173.
[48] Nguyen M.H., Nguyen T.-A., Ly H.-B. Ensemble XGBoost schemes for improved compressive strength prediction of UHPC. Elsevier. p. 105062.
[49] Sun Z, Li Y, Yang Y, Su L, Xie S. Splitting tensile strength of basalt fiber reinforced coral aggregate concrete: optimized XGBoost models and experimental validation. Constr Build Mater 2024;416:135133.
[50] Usta M., Pujol S., 445B A.S., Puranam A., Song C., Wang Y. ACI 445B shear wall database. 2017.
[51] Ang Alfredo HS, Amin M. Safety factors and probability in structural design. J Struct Div 1969;95:1389–405.
[52] Goldberg DE. Genetic algorithms in search, optimization and machine learning. Addison-Wesley Longman Publishing Co., Inc; 1989.
[53] Taguchi Y, Hino H, Kameyama K. Pre-training acquisition functions by deep reinforcement learning for fixed budget active learning. Neural Process Lett 2021; 53:1945–62.
[54] Zhihui X.Z., Lin, Junyou L., Shuai L., Deheng Y. Pretraining in deep reinforcement learning: a survey. arXiv preprint arXiv:221103959, 2022. 2022.