

# Deep reinforcement learning-based critical element identification and demolition planning of frame structures

Shaojun ZHU<sup>a,b,c</sup>, Makoto OHSAKI<sup>b</sup>, Kazuki HAYASHI<sup>b</sup>, Shaohan ZONG<sup>a\*</sup>, Xiaonong GUO<sup>a</sup>

<sup>a</sup> College of Civil Engineering, Tongji University, Shanghai 200092, China

<sup>b</sup> Department of Architecture and Architectural Engineering, Kyoto University, Kyoto 615-8540, Japan

<sup>c</sup> Tongji Lvjian Co., Ltd., Shanghai 200092, China

\*Corresponding author. E-mail: shzong@tongji.edu.cn

© Higher Education Press 2022

**ABSTRACT** This paper proposes a framework for critical element identification and demolition planning of frame structures. Innovative quantitative indices considering the severity of the ultimate collapse scenario are proposed using reinforcement learning and graph embedding. The action is defined as removing an element, and the state is described by integrating the joint and element features into a comprehensive feature vector for each element. By establishing the policy network, the agent outputs the  $Q$  value for each action after observing the state. Through numerical examples, it is confirmed that the trained agent can provide an accurate estimation of the  $Q$  values, and handle problems with different action spaces owing to utilization of graph embedding. Besides, different behaviors can be learned by varying hyperparameters in the reward function. By comparing the proposed method and the conventional sensitivity index-based methods, it is demonstrated that the computational cost is considerably reduced because the reinforcement learning model is trained offline. Besides, it is proved that the  $Q$  values produced by the reinforcement learning agent can make up for the deficiencies of existing indices, and can be directly used as the quantitative index for the decision-making for determining the most expected collapse scenario, i.e., the sequence of element removals.

**KEYWORDS** progressive collapse, alternate load path, demolition planning, reinforcement learning, graph embedding

## 1 Introduction

In the process of designing building frames, safety against various loads, including self-weight, live loads, seismic excitation, and wind loads, is considered. Furthermore, if more critical events leading to total collapse are to be considered, safety against progressive collapse may be another important aspect of the design process, as it significantly threatens the safety of human lives and properties. Thus, the anti-progressive collapse design has been extensively studied in the recent two decades, where the alternate load path (ALP) method [1] is a well-accepted approach to evaluate the redundancy of the structure under local failure [2–5]. Specifically, the ALP method modifies the structural system assuming the loss

of one or multiple structural elements. The critical elements, after whose failure the ALP can hardly be established, can be quantitatively identified by sensitivity indices that characterize the effect of loss of the element on the total strength and difficulty in the internal force redistribution of the frame. Most commonly-used sensitivity indices are determinate ones based on the bearing capacity [7–10], deformation [11], dimensionless total damage [12], or energy [13,14]. Another typical engineering practice sharing the same mechanism is demolition planning (DP) [15–17], which aims at safely demolishing the whole structure by eliminating structural elements using controlled explosions or mechanical demolition [18]. Apart from the internal force redistribution severity, the DP also takes the cost and efficiency into consideration. Correspondingly, Isobe [19] proposed the key element index to estimate the contribution of the

structural elements to overall collapse for a successful demolition.

However, two main difficulties arise when the sensitivity indices mentioned above are used to conduct critical element identification (CEI) or DP. The first is that the calculation of indices requires structural analysis for the damaged structure corresponding to each scenario of element removal at each step of collapse or demolishing analysis. Hence, the computational cost can be considerable if all possible scenarios are traversed, as structures with high redundancy will collapse under multiple-element loss instead of single-element loss. The other difficulty is that, as these indices cannot consider the outcome of the ultimate collapse and the sequence of the element loss (or the collapse process), they cannot be directly applied to make decisions for determining the most expected collapse scenario [20], and human inference or optimization methods are necessary, which can be laborious.

Regarding the first difficulty, the methods called exact or approximate reanalysis can be utilized instead of carrying out analysis for the frame corresponding to each removal scenario. Ohsaki [21] proposed an exact reanalysis method for truss structures, and complicated matrix operations to calculate the inverse stiffness matrix of the modified structure are avoided in the process of topology optimization. Makode et al. [22] used the virtual distortion method and reduced the order of matrix equations for the reanalysis of rigidly jointed frames. However, some approximation errors inherently exist for rigidly jointed frames.

As for the second difficulty, although stochastic sensitivity indices considering the failure probability [23,24], risk [25,26], and reliability [27,28] are proposed and are of theoretical significance, they are not friendly to practical engineers in the initial design process since additional computational cost of probability analysis is brought in. Nonetheless, it is evident that a certain relationship exists between the element(s)-loss scenarios and the properties of the damaged frames because the structural properties are determined by the locations and properties of members, and they can be intuitively estimated by the experienced structural designers and engineers. Besides, different element(s)-loss scenarios can be characterized by different Markov Decision Processes (MDPs) [29], where the sequence of element loss and the outcome of the final collapse can be naturally considered. Hence, the machine learning techniques are promising in utilizing the existing data/experience to reduce the computational cost [30] for CEI and also for optimizing the DP, as well as resolving the shortcomings of existing sensitivity indices.

In recent years, supervised learning (SL) and reinforcement learning (RL) have been extensively applied in the field of structural engineering, as recent progress is

summarized in literature [31,32]. Specifically, SL can effectively handle static regression problems and learn experience from the training data by means of establishing neural networks. Hence, problems in structural engineering that involves high non-linearity and large computational cost can be solved by establishing surrogate models based on SL. For instance, Zhu et al. [33] utilized the artificial neural networks and the support vector machine to estimate the non-linear buckling capacity of imperfect reticulated structures, and the computational time of non-linear buckling analysis is significantly reduced. Besides, problems related to time history can also be handled by SL. Xue et al. [34] established a surrogate model using the convolutional neural networks to predict the time history response of transmission towers under complex wind inputs. On the other hand, RL deals with problems that involve interaction between the task and the environment by characterizing the task into MDPs, e.g., optimizing the structure with respect to its mechanical performance, arranging members to form a reasonable structure, *etc.* With the outstanding regression performance of neural networks, deep neural networks (DNNs), where multiple neural networks are incorporated, are extensively used [35–40]. Some researchers also combined DNNs with RL to form deep RL. For example, Li et al. [41] developed a deep RL-based shape optimizer using the recurrent neural networks, which is able to provide optimal shapes for wind-sensitive buildings with low computational cost. NP-hard combinatorial optimization problems can also be well handled using deep RL. Hayashi and Ohsaki [42] proposed a topology optimization method for 2D truss structures where an RL agent instead of commonly-used iterative algorithms is used to eliminate the members. Zhu et al. [43] also trained an RL agent that is able to generate machine-specified ground structures that are random yet reasonable for topology optimization. It is notable that the graph embedding (GE) technique [44,45] was adopted in studies [42,43] so that the trained agent can be applied to different-sized problems without re-training. The GE technique better fits for extracting features of discrete structures, including trusses, building frames, and reticulated shells, as the joints and elements can be abstracted as nodes and edges in a graph. By establishing fully-connected neural networks, the global feature of the whole graph, namely the whole structure, can be integrated into nodes or edges as requested. A more specific description of the GE technique can be found in Subsections 3.2 and 3.3 in this paper.

Since RL has successfully been adopted to distinguish vulnerable joints in large-scale cyber systems [46] and identify key players in complex networks [47], this paper proposes a method incorporating RL and GE for reducing the computational cost of reanalysis in the CEI and the DP process. The main motivation of the proposed method

is to make up for the two shortcomings of sensitivity indices mentioned above. The paper is organized as follows: Section 2 describes two tasks and the corresponding aims in detail. Section 3 introduces the key points of the proposed RL framework, including the state, action, reward function, the policy network, and the learning method. Section 4 exhibits three numerical examples illustrating the application and advantage of the proposed method for building frames. Section 5 summarizes the conclusions obtained in this paper.

## 2 Task and aim

### 2.1 Task 1: CEI against progressive collapse

The ALP method removes one or several structural elements to simulate the damage of the structure and evaluates the structural robustness by calculating the element sensitivity index defined as

$$\gamma_i = \left| 1 - \frac{\xi_i}{\xi_0} \right|, \quad (1)$$

where  $\gamma_i$  is the sensitivity index of the  $i$ th element ( $i = 1, 2, \dots, n_e$ );  $n_e$  is the total number of removable elements;  $\xi_0$  and  $\xi_i$  are the global responses of the original structure and the remaining structure with the  $i$ th element removed, respectively. The global response can be the maximum stress ratio [8], total strain energy [14], determinant of the stiffness matrix [48], etc. If the element response is to be used, Eq. (1) is rewritten as

$$\gamma_i = \max_j \left| 1 - \frac{\xi_{j,i}}{\xi_{j,0}} \right|, \quad (j \neq i \text{ and } i \in \Omega_r), \quad (2)$$

where the subscript  $j$  indicates the response corresponds to the  $j$ th element, and  $\Omega_r$  is the set of indices of the remaining elements that can be removed.

Generally speaking, critical elements can be defined as those with high sensitivity indices since the loss of these elements leads to a more significant internal force redistribution than that of other low sensitivity elements. Nonetheless, the sensitivity index defined in Eq. (1) or (2) cannot reflect the importance of the element considering the ultimate collapse scenario [20]. Besides, it is indicated that progressive collapse may not be triggered even if considerable initial damage occurs [49]. Therefore, the importance of each element should be evaluated when the total collapse state is achieved after several steps of the element removal process. That is to say, the impact of multiple-element loss also needs to be investigated, although the number of removed elements should be limited because total collapse with a smaller number of elements indicates higher importance of the removed elements. Thus, the sensitivity index is proposed to be

evaluated at each step of the sequential removal process as follows [19]:

$$\gamma_{i,t} = \left| 1 - \frac{\xi_{i,t}}{\xi_{0,t}^{(i)}} \right|, \quad (3)$$

where  $\gamma_{i,t}$  is the sensitivity index of the  $i$ th element at the  $t$ th element removal step;  $\xi_{0,t}^{(i)}$  and  $\xi_{i,t}$  are the global responses of the structure at the  $t$ th element removal step before and after the loss of the  $i$ th element, respectively. Similarly, if the element response is used, Eq. (2) is rewritten as

$$\gamma_{i,t} = \max_j \left| 1 - \frac{\xi_{j,i,t}}{\xi_{j,0,t}^{(i)}} \right|, \quad (j \neq i \text{ and } i \in \Omega_r). \quad (4)$$

Nonetheless, it is evident that  $\gamma_{i,t}$  is only related to the responses at time steps  $t$  and  $t-1$ , and it cannot evaluate the impact of the ultimate collapse scenario.

Therefore, a framework for CEI against progressive collapse based on a quantitative index is expected to have the following properties:

- 1) sequence of removal of multiple elements is considered;
- 2) the sensitivity indices of the remaining elements can be computed with a low computational cost;
- 3) the ratio of the collapsed part to the whole structure is considered to quantify the severity of the ultimate collapse scenario;
- 4) the number of elements removed before the ultimate collapse is incorporated.

### 2.2 Task 2: Structural DP

Conventional methods of the demolition of building structures often involve nonexplosive demolition agents, which are costly and time-consuming [19]. However, the DP of controlled explosives or mechanical demolition requires a certain level of engineering expertise. Demolition of a structure expects an overall collapse, i.e., the ratio of the collapsed part to the overall structure should account for a higher percentage than the CEI against progressive collapse. Besides, a successful demolition should also take safety and cost into consideration. For example, it can be dangerous to remove a ground-floor column in a frame at the first step. Nonetheless, it is notable that the definition of safety and cost should be subjective.

Therefore, a framework for DP of structures based on a model-dependent index is expected to be proposed. The index should have the following properties:

- 1) severity of the ultimate overall collapse scenario is considered;
- 2) safety and cost of the removal of elements are considered;

3) Different importance of the two aspects mentioned above, specified based on engineering judgment, can be incorporated.

Notably, the indices that satisfy the requirements proposed in Subsections 2.1 and 2.2 exist and are of significance to the corresponding tasks, because theoretically there should be a deterministic most disadvantageous ultimate collapse scenario for each task. However, they cannot be obtained based on existing methods, and Section 3 proposes a feasible framework to resolve this problem using RL.

### 3 RL framework

Both tasks described in Section 2 involve element removal, which can be regarded as an interaction between an agent and the environment in the framework of RL. Besides, since the scenario of removing multiple elements is considered in both tasks, the number of possible states of the environment, which can be defined as the combination of removed/existing elements, becomes so large that it is computationally expensive to prepare the labeled training data for SL. However, scenarios of removing process of multiple elements can be formulated as MDPs in RL. Therefore, RL is selected as the technique to obtain approximate optimal sequences of element removal for both CEI and DP in a single framework using different hyperparameters.

Generally, the interaction between an RL agent and the environment is performed through the following steps:

1) in the  $t$ th step, the agent observes the current state of the environment  $s_t$ ;

2) by observing  $s_t$ , the agent takes action  $a_t$  according to a policy  $\pi$  which is usually established using the neural networks, namely

$$a_t = \pi(s_t); \quad (5)$$

3) the state transfers from  $s_t$  to  $s_{t+1}$  due to the action  $a_t$  and the agent receives a reward  $r_{t+1}$  from the environment;

4) repeat from 1) with the state being  $s_{t+1}$ .

The training/learning of the agent is aimed at maximizing the cumulative reward by updating the policy  $\pi$  based on the sequence of data ( $s_t, a_t, r_{t+1}, s_{t+1}$ ). Various learning methods have been proposed, including Q-learning [50], REINFORCE [51], PPO [52], etc., corresponding to the value-based, policy-based, and the actor-critic framework, respectively. Therefore, the key components in a typical process of RL include the action, state, reward function, policy (network), and learning method.

#### 3.1 Action

Since both tasks 1 and 2 in Section 2 involve element

removal, the action of the RL problem is determined as element selection. Hence, the action space is

$$\mathcal{A} = \{1 \ 2 \ 3 \ \cdots \ n_e\}. \quad (6)$$

#### 3.2 State

The structural features of frames can generally be classified into two types, i.e., the joint and element features. However, not all structural features need to be integrated into the state for the agent to observe. Hence, the input data size can be decreased to reduce the computational cost for training.

##### 1) Joint feature vector

Considering the aims of the two tasks, the necessary joint features include the joint coordinates and the support condition. Therefore, the feature vector of the  $i$ th joint  $\mathbf{v}_i$  ( $i = 1, 2, 3, \dots, n_n$ ), where  $n_n$  is the number of joints in the structure, is constructed as

$$\mathbf{v}_i = \{\bar{x}_i \ \bar{y}_i \ \bar{d}_i \ F_i\}^T, \quad (7)$$

where  $\bar{x}_i$  and  $\bar{y}_i$  are the normalized horizontal and vertical coordinates of the  $i$ th joint in the  $x$  and  $y$  directions, respectively, calculated by

$$\begin{cases} \bar{x}_i = \frac{x_i - \min(\mathbf{x})}{\max(\Delta x, \Delta y)}, \\ \bar{y}_i = \frac{y_i - \min(\mathbf{y})}{\max(\Delta x, \Delta y)}, \end{cases} \quad (8)$$

where  $x_i$  and  $y_i$  are the actual coordinates of the  $i$ th joint in the  $x$  and  $y$  directions, respectively;  $\mathbf{x}$  and  $\mathbf{y}$  are the sets of horizontal and vertical joint coordinates, namely

$$\begin{cases} \mathbf{x} = \{x_1 \ x_2 \ x_3 \ \cdots \ x_{n_n}\}, \\ \mathbf{y} = \{y_1 \ y_2 \ y_3 \ \cdots \ y_{n_n}\}. \end{cases} \quad (9)$$

where  $\Delta x$  and  $\Delta y$  are the range of the horizontal and vertical joint coordinates, respectively, calculated by

$$\begin{cases} \Delta x = \max(\mathbf{x}) - \min(\mathbf{x}), \\ \Delta y = \max(\mathbf{y}) - \min(\mathbf{y}). \end{cases} \quad (10)$$

$\bar{d}_i$  is the normalized distance between the  $i$ th joint and the nearest support, calculated by the following equation if the joint number of the nearest support is supposed to be  $k$ :

$$\bar{d}_i = \sqrt{(\bar{x}_i - \bar{x}_k)^2 + (\bar{y}_i - \bar{y}_k)^2}. \quad (11)$$

$F_i$  is the downward concentrated load at the  $i$ th joint. Note that  $F_i$  can be omitted if there are no concentrated loads at joints, i.e., the size of vector  $\mathbf{v}_i$  is reduced to 3.

## 2) Element feature vector

Considering the aims of the two tasks, the necessary element features include the geometric dimensions, material properties, existence/nonexistence, and the structural response. Since the target of incorporating RL involves a reduction in the computational cost, the calculation of the structural response should avoid multiple times of reanalysis. Therefore, the feature vector of the  $j$ th element  $\mathbf{m}_j$  ( $j = 1, 2, 3, \dots, n_e$ ) is constructed as

$$\mathbf{m}_j = \{l_j \quad \text{Sel}_j \quad q_j \quad f_{y,j} \quad A_j \quad I_{z,j} \quad C_j \quad R_{C,j}\}^T, \quad (12)$$

where the subscript  $j$  indicates the value of the  $j$ th element;  $\text{Sel}_j = 0$  and 1 indicate the nonexistence and existence, respectively, of the element;  $l_j$ ,  $A_j$ , and  $I_{z,j}$  are the length, cross-sectional area, and moment of inertia about the strong axis, respectively;  $f_{y,j}$  is the yield strength of the material;  $C_j$  is the strain energy;  $R_{C,j}$  is the strain energy ratio, calculated by

$$R_{C,j} = \frac{C_j}{C_{\max}}, \quad (13)$$

where  $C_{\max}$  is the maximum strain energy of a single element out of the  $n_e$  elements;  $q_j$  is the intensity of the downward distributed load per unit length of the beam. Note that  $q_j$  can be omitted if there is no distributed load on the beam, i.e., the size of vector  $\mathbf{m}_j$  is reduced to 7.

## 3) Comprehensive feature vector

Here we notice that the size of the action space  $\mathcal{A}$  equals the number of elements, and we expect a quantitative index for each element. Therefore, it is reasonable to integrate the joint and element features into elements as the comprehensive feature. Let  $\boldsymbol{\mu}_j$  denote the comprehensive feature vector of the  $j$ th element with a size of  $n_f$ . Note that  $n_f$  is a hyperparameter that should be greater than the sizes of  $\mathbf{v}_i$  and  $\mathbf{m}_j$ .

The edge-embedding technique in GE [45] is used to calculate  $\boldsymbol{\mu}_j$  through an iteration process as

$$\boldsymbol{\mu}_j^{(0)} = \mathbf{0}, \quad (14)$$

$$\boldsymbol{\mu}_j^{(T+1)} = \text{ReLU}[\mathbf{h}_1 + \mathbf{h}_2 + \mathbf{h}_3^{(T)} + \mathbf{h}_4^{(T)}], \quad (15)$$

where  $\boldsymbol{\mu}_j^{(T)}$  is the comprehensive feature vector of the  $j$ th element at the  $T$ th iteration. ReLU is an activation function defined for a real value  $x$  as

$$\text{ReLU}(x) = \max\{0, x\}. \quad (16)$$

When the ReLU function is applied to a matrix, we assume the element-wise application of Eq. (16) for simplicity.  $\mathbf{h}_1$ ,  $\mathbf{h}_2$ ,  $\mathbf{h}_3^{(T)}$ , and  $\mathbf{h}_4^{(T)}$  are intermediate vectors calculated as

$$\mathbf{h}_1 = \boldsymbol{\theta}_1 \mathbf{m}_j, \quad (17)$$

$$\mathbf{h}_2 = \boldsymbol{\theta}_3 \sum_{i=1}^2 \text{ReLU}(\boldsymbol{\theta}_2 \mathbf{v}_{ji}), \quad (18)$$

$$\mathbf{h}_3^{(T)} = \boldsymbol{\theta}_4 \boldsymbol{\mu}_j^{(T)}, \quad (19)$$

$$\mathbf{h}_4^{(T)} = \boldsymbol{\theta}_6 \sum_{i=1}^2 \text{ReLU} \left[ \boldsymbol{\theta}_5 \sum_{k \in \Phi_{ji}} \boldsymbol{\mu}_k^{(T)} \right], \quad (20)$$

where  $\mathbf{v}_{ji}$  is the joint feature vector of the  $i$ th ( $i \in \{1, 2\}$ ) end of the  $j$ th element;  $\Phi_{ji}$  is the set of element indices connected to the  $i$ th end of the  $j$ th element except for the  $j$ th element itself;  $\boldsymbol{\theta}_1$ – $\boldsymbol{\theta}_6$  are the weight matrices of the fully-connected neural network layers in the GE network for the comprehensive feature vector, whose sizes are tabulated in Table 1, where  $n_v$  and  $n_m$  are the sizes of the joint and element feature vectors, respectively. A simple illustration of the iteration process of Eqs. (14) and (15) consisting of 6 nodes and 5 edges can be found in Fig. 1. It is notable that the embedding indicated by the arrows in Fig. 1(b) is realized by weight matrices  $\boldsymbol{\theta}_1$ – $\boldsymbol{\theta}_6$ .

Let  $T_{\max}$  denote the maximum number of iterations in the embedding process, and let  $\hat{\boldsymbol{\mu}}$  and  $\boldsymbol{\mu}_j$  denote  $\hat{\boldsymbol{\mu}}^{(T_{\max})}$  and  $\boldsymbol{\mu}_j^{(T_{\max})}$ , respectively, for simplicity, where  $\hat{\boldsymbol{\mu}}^{(T)} = \{\boldsymbol{\mu}_1^{(T)} \quad \boldsymbol{\mu}_2^{(T)} \quad \boldsymbol{\mu}_3^{(T)} \quad \dots \quad \boldsymbol{\mu}_{n_e}^{(T)}\}$  is the comprehensive feature matrix at the  $T$ th iteration. Obviously, the larger  $T_{\max}$  is, the features of further joints and elements can be integrated into the comprehensive feature vector of a single element; however, the computational cost will also increase. Hence,  $T_{\max}$  is also a hyperparameter to be tuned.

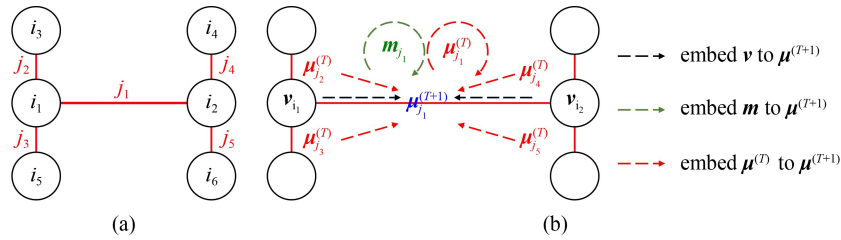
By adequately selecting the values of  $n_f$  and  $T_{\max}$ , the elements in  $\hat{\boldsymbol{\mu}}$  are expected to contain the joint and element features of the whole structure. Thus, it is reasonable to adopt  $\hat{\boldsymbol{\mu}}$  to describe the state of the structure.

## 3.3 Policy network

Here we state again that tasks of CEI and DP aim at exploring the deterministic set of elements that yield the

**Table 1** Sizes of the neural network layers

| layers for the comprehensive feature vector |                  | layers for the DQN      |                  |
|---|------------------|-------------------------|------------------|
| weight matrix                               | size             | weight matrix           | size             |
| $\boldsymbol{\theta}_1$                     | $n_f \times n_m$ | $\boldsymbol{\theta}_7$ | $n_f \times n_f$ |
| $\boldsymbol{\theta}_2$                     | $n_f \times n_v$ | $\boldsymbol{\theta}_8$ | $n_f \times n_f$ |
| $\boldsymbol{\theta}_3$                     | $n_f \times n_f$ | $\boldsymbol{\theta}_9$ | $1 \times 2n_f$  |
| $\boldsymbol{\theta}_4$                     | $n_f \times n_f$ | –                       | –                |
| $\boldsymbol{\theta}_5$                     | $n_f \times n_f$ | –                       | –                |
| $\boldsymbol{\theta}_6$                     | $n_f \times n_f$ | –                       | –                |



**Fig. 1** Illustration of edge-embedding. (a) Numbering of nodes and edges; (b) iteration process.

most severe ultimate collapse scenario. Hence, the value-based framework of RL, which establishes a deterministic quantitative index for the value of actions, is more suitable. Based on the state  $\hat{\mu}$ , the quantitative index  $Q_j$ , representing the value of the  $j$ th action, is expected to be estimated by the following deep  $Q$  network (DQN):

$$Q_j = \tilde{\pi}(\mu_j) = \theta_7 \text{ReLU} \left( \theta_8 \mu_j; \theta_9 \sum_{j=1}^{n_e} \mu_j \right), \quad (21)$$

where  $\tilde{\pi}(\cdot)$  indicates the output of the DQN; operator  $(;\cdot)$  denotes that two vectors are concatenated in the column direction;  $\theta_7$ – $\theta_9$  are the weight matrices of the neural network layers in the DQN, whose sizes are tabulated in Table 1. Notably, the structure of the DQN is determined by the property of the RL task, as we are expecting the output, i.e., the quantitative indices for elements, with a size of  $n_e \times 1$ . It can be seen from Table 1 that the sizes of the weight matrices are independent of the number of joints and elements, i.e.,  $n_n$  and  $n_e$ , by incorporating the edge-embedding technique. Thus, the trained agent is expected to handle different-sized problems, i.e., frames with different numbers of joints and elements.

Note that the  $Q$  value is available for each action, and the final action  $a$  is yet to be selected according to the following policy network:

$$a = \pi(Q) = \pi[\tilde{\pi}(\hat{\mu})], \quad (22)$$

where  $Q = \{Q_1 \ Q_2 \ Q_3 \ \cdots \ Q_{n_e}\}$  is the set of  $Q$  values of actions;  $\pi$  is the policy of selecting an action based on  $Q$ , which will be introduced in Subsection 3.5.

### 3.4 Reward function

If the following criterion is satisfied for every beam element in the structure, the structure can be regarded as satisfying the anti-progressive collapse requirement [53,54]:

$$\varphi_b \leq \varphi_{\text{lim}} = 0.0213 - 0.00012h, \quad (23)$$

where  $\varphi_b$  is the plastic rotation of beam element (rad) and  $\varphi_{\text{lim}}$  is the corresponding upper limit (rad);  $h$  is the height of the cross-section of the beam (cm).

Let  $\lambda_i$  ( $> 0$ ) ( $i = 1, 2, 3, 4$ ) denote the hyperparameters in the reward function, and define an episode as the process

of removing elements in a structure until the anti-progressive collapse requirement is violated. In the  $t$ th step, the reward  $r_{t+1}$  of action  $a_t$  in an episode is determined as follows:

1) If there is an isolated part that is not connected to any support, the structural analysis will not be performed, and the reward of the current action is

$$r_{t+1} = \lambda_0 \lambda_1 R_{b,0}, \quad (24)$$

where  $R_{b,0}$  is a ratio defined as

$$R_{b,0} = \frac{n_{b,0}}{n_b}, \quad (25)$$

where  $n_{b,0}$  is the number of isolated beams, and  $n_b$  is the total number of beams. Note that  $\lambda_0$  can be taken as 1 to indicate the isolated part is equivalently regarded as collapsed;  $\lambda_0$  can also be taken as a negative value to indicate the isolated part is not favored and prevent locally isolated parts that can often exist in the training process. The episode will be terminated at the current step;

2) If there is no isolated part in the structure, conduct the structural analysis to check criterion Eq. (23) and calculate the strain energy for  $m_j$ . Note that the structural analysis is carried out only once at each step. Then, the reward of the current action is

$$r_{t+1} = \begin{cases} -\lambda_2 \text{Re}^{-\lambda_3 k_{\text{sen},a}} - \lambda_4, & \text{if requirement is satisfied,} \\ -\lambda_2 \text{Re}^{-\lambda_3 k_{\text{sen},a}} - \lambda_4 + \lambda_1 R_{b,e}, & \text{if requirement is violated,} \end{cases} \quad (26)$$

where  $R$  is the reduction factor of the cost of removing an element, which can vary with the internal force level of the element as described in the numerical examples;  $k_{\text{sen},a}$  is the strain energy sensitivity index of the action in the current step defined as

$$k_{\text{sen},a} = \max_j \frac{C_{j,a} - C_{j,0}}{C_{j,0}}, \quad (j \neq a \text{ and } j \in \Omega_t), \quad (27)$$

where  $C_{j,a}$  and  $C_{j,0}$  are the strain energy of the  $j$ th element after and before execution of the action, i.e., removal of element  $a$ , respectively.  $R_{b,e}$  is a ratio defined as

$$R_{b,e} = \frac{n_{b,e}}{n_b}, \quad (28)$$

where  $n_{b,e}$  is the number of beams violating Eq. (23). The

episode will be terminated at the current step if the following requirement is satisfied:

$$\begin{cases} R_{b,e} \geq \lambda_R, & \text{for } \lambda_R > 0, \\ R_{b,e} > \lambda_R, & \text{for } \lambda_R = 0, \end{cases} \quad (29)$$

where  $\lambda_R$  is a hyperparameter indicating the lower limit of  $R_{b,e}$  to terminate the episode. Note that different values of  $\lambda_R$  are used for CEI and DP.

Notably, the aims described in Section 2 have been considered by the reward function. The positive rewards in Eqs. (24) and (26) incorporate the severity of the ultimate collapse. As the term  $-\lambda_4$  in Eq. (26) can represent the constant cost of demolishing an element, the term  $-\lambda_2 Re^{-\lambda_3 k_{sen,a}}$  can serve as the importance of the element, since  $k_{sen,a}$  reflects the severity of the internal force redistribution after the loss of the element. In addition, hyperparameter  $\lambda_R$  is used to adjust the requirement of the severity of the ultimate collapse to terminate the episode. A small value close to 0 is given for Task 1, and a large value close to 1 is given for Task 2. Hence, agents with different behaviors corresponding to the two tasks can be trained with a unified reward function.

### 3.5 Learning method

The typical value-based learning method Q-learning [50] is used to update the network parameters. The general aim of Q-learning is to minimize the difference between the estimated  $Q$  value and the observed reward, i.e., the  $Q$  value at the  $t$ th step is updated as

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[ r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t) \right], \quad (30)$$

where  $\alpha$  is the learning rate that controls the variation amplitude of the network parameters and is closely associated with the convergence of the training process;  $\gamma$  is the discount factor on the future reward within the range of [0,1]. Notably, the value of  $\gamma$  indicates the percentage of future reward to be included in the estimated  $Q$  value produced by the trained agent, where  $\gamma = 0$  expects the final  $Q$  value to be close to the instant reward and  $\gamma = 1$  expects the final  $Q$  value to be close to the sum of the instant and future rewards. Since the  $Q$  value is the output of the DQN  $\tilde{\pi}$ , the update process of Eq. (30) can be realized by solving the following optimization problem:

$$\begin{cases} \text{find } \boldsymbol{\theta} \\ \text{min } L(\boldsymbol{\theta}) = \frac{1}{2} \left[ r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a'; \tilde{\boldsymbol{\theta}}) - Q(s_t, a_t; \boldsymbol{\theta}) \right]^2, \end{cases} \quad (31)$$

where  $\boldsymbol{\theta} = \{\theta_1, \theta_2, \dots, \theta_9\}$  is the set of network parameters,

and  $\tilde{\boldsymbol{\theta}}$  is the most recent set of network parameters, i.e.,  $\tilde{\boldsymbol{\theta}}$  contains the values in  $\boldsymbol{\theta}$  at the last optimization step, which has been proved to facilitate the convergence [55];  $L(\boldsymbol{\theta})$  is the loss function describing the difference between the  $Q$  value and the actual reward.

Optimization problem (31) can be solved using the typical error back-propagation method [56], and the RMSprop optimizer [57] which can adjust the gradient adaptively according to the momentum is recommended. However, solving problem Eq. (31) using the batch containing every sampled record of  $(s_t, a_t, r_{t+1}, s_{t+1})$  will be computationally expensive. Hence, the prioritized experience replay (PER) method [58] is adopted to sample a batch of records with higher priority from the set of records  $D$ , which is also called the experience replay buffer. Let  $n_D$  denote the capacity of  $D$ , i.e., the maximum number of records. The size of the batch  $n_{ba}$  is generally smaller than  $n_D$ . The priority of a record is evaluated according to the temporal-difference error  $\delta$ , i.e., the loss function  $L$ , since the following relation holds:

$$\delta(s_t, a_t, r_{t+1}, s_{t+1}) = r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t) = \sqrt{2L}. \quad (32)$$

A record with a higher  $\delta$  indicates a greater error between the  $Q$  value and the actual reward; thus, the record should have a higher priority to be learned more frequently. Besides, the latest record added to  $D$  will be assigned the highest priority to ensure that every record can be learned at least once. Hence, the adoption of PER can accelerate the training by utilizing the records that induce more changes in the trainable parameters.

In order to achieve the best performance, the agent trained by Q-learning can take actions according to the greedy policy [29]:

$$a = \pi(\boldsymbol{Q}) = \operatorname{argmax}_{a' \in \Omega_r} Q_{a'}. \quad (33)$$

However, the greedy policy can be poor at exploring unknown states for better policies if used in the training process. This is because the records tend to be repetitive once the policy network converges at a local optimum. In other words, only the  $Q$  values of the optimal actions will be accurate. Since a more accurate quantitative index is preferred, the epsilon-greedy policy [29] is adopted in the training process for sampling the actions:

$$a = \pi(\boldsymbol{Q}) = \begin{cases} \operatorname{argmax}_{a' \in \Omega_r} Q_{a'}, & p = 1 - \varepsilon, \\ \text{random } a' \in \Omega_r, & p = \varepsilon, \end{cases} \quad (34)$$

where  $p$  is the possibility of adopting the policy, and  $\varepsilon$  is the exploration rate within the range of [0, 1]. Note that the value of  $\varepsilon$  can be either a constant or a variable depending on the number of episodes.

From the update method in Eq. (30), the  $Q$  value of the action  $a_t$  at state  $s_t$ , i.e.,  $Q(s_t, a_t)$  will converge at

$$Q(s_t, a_t) \rightarrow r_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a'). \quad (35)$$

As discussed in Section 3.4, the reward function has fulfilled all aims of the tasks; therefore, the  $Q$  values are expected to serve as an improved sensitivity index for the tasks.

The flow chart of the proposed deep RL framework is plotted in Fig. 2.

## 4 Numerical examples

### 4.1 Numerical model

A  $4 \times 5$  planar steel frame shown in Fig. 3 is used to train the agent as a numerical example. The structural information is shown as follows:

- 1) span:  $4 \times 8 = 32$  m;
- 2) height:  $5 \times 4 = 20$  m;
- 3) distributed beam load intensity:  $q = 32$  kN/m (downward);
- 4) support condition: fixed support at the ground joints;
- 5) elastic modulus of the steel:  $E = 2.06 \times 10^5$  MPa;
- 6) yield strength of the steel:  $f_y = 235$  MPa.

Note that only the vertical load is considered as the load case according to the anti-progressive collapse design guides/codes [53,59]; besides, only the columns can be

removed, i.e.,  $\Omega_r$  is the set of remaining columns. The cross-sections of the structural elements are specified by circled numbers ①–④, where all the beams share the same cross-section ①, and the columns in the same story have the same cross-section specified on their left.

The numerical model of the structure is established in the general finite element analysis software package ANSYS [60]. The BEAM188 element is used to model the beams and columns. Both geometric and material non-linearities are considered, and the ideal elastic-plastic constitutive model is used for the steel material. The arc-length method [61,62] is used to perform the non-linear analysis.

### 4.2 Training

The training of the agent is conducted in Python 3.8.8 environment, and the interaction between Python and ANSYS is realized using the PyMAPDL library [63]. The hyperparameters of both Tasks 1 and 2 are identical except for those in the reward function, as tabulated in Table 2, where  $n_{ep}$  indicates the total number of episodes and  $e$  is the episode number. Hyperparameters regarding the GE network, i.e.,  $T_{max}$  and  $n_f$ , are selected according to the recommendations in literature [42,43];  $\gamma$  is selected as 1 in order to fully consider the influence of the future states;  $n_{ep}$ ,  $n_{ba}$ , and  $\alpha$  are selected based on trial and error. Note that two different values of  $\lambda_4$  are selected for Task 2 to represent different importance of safety and cost. The two values, i.e., 10 and 5, are denoted as the high-cost

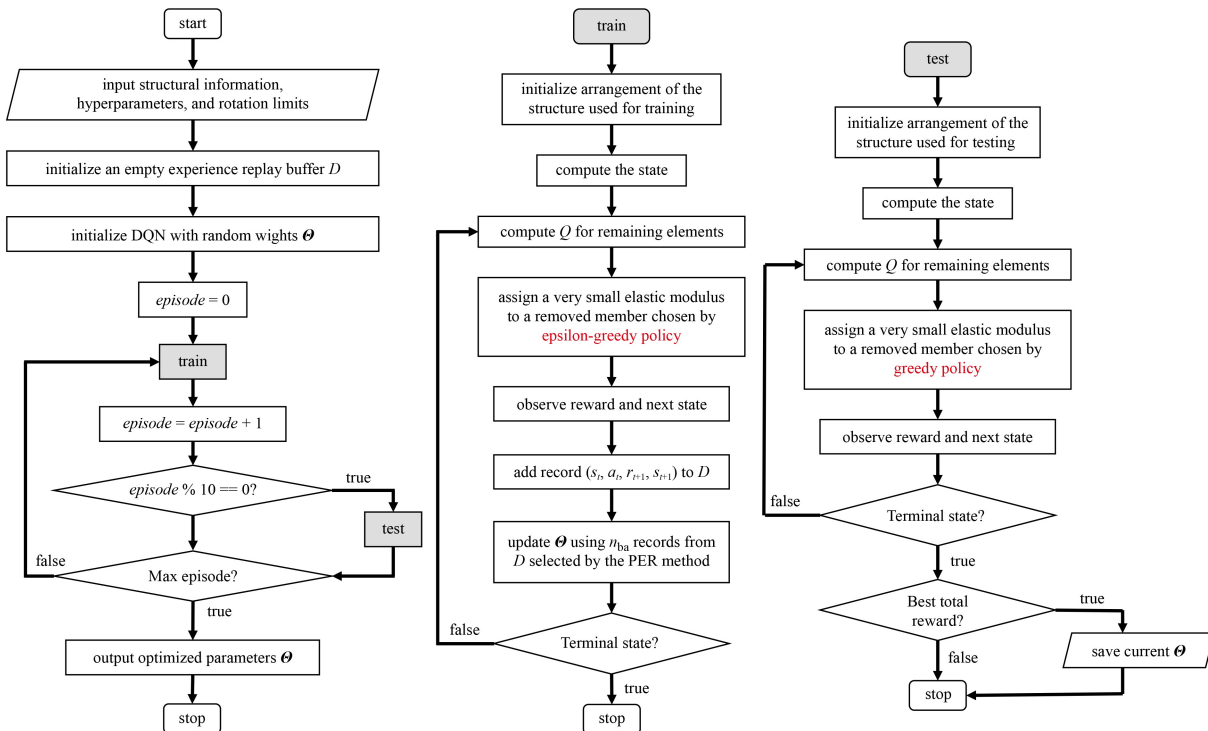


Fig. 2 Flow chart of the proposed RL framework.

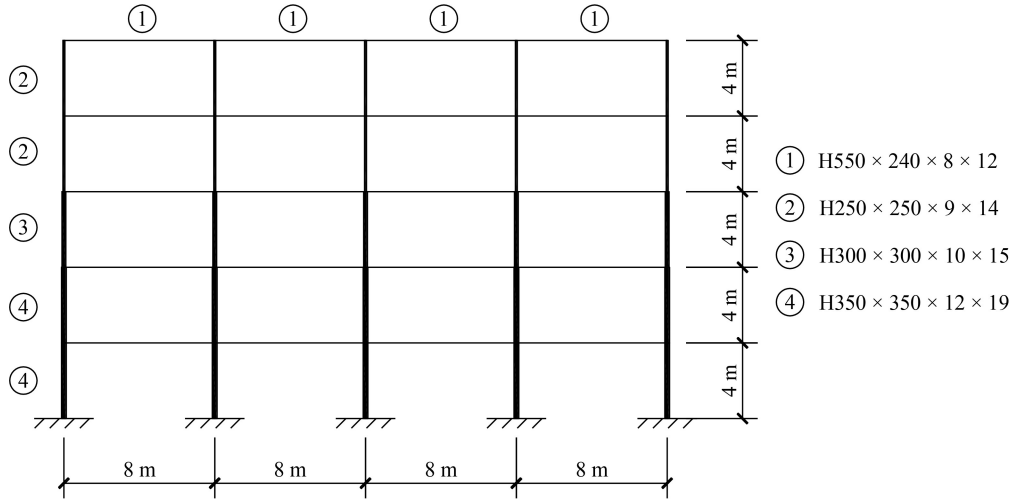


Fig. 3 A 4 × 5 planar frame.

Table 2 Hyperparameters of the numerical example

| general parameters |   | reward parameters of Task 1 |                    | reward parameters of Task 2    |                           |
|--------------------|---|-----------------------------|--------------------|--------------------------------|---------------------------|
| symbol             | value   | symbol                      | value              | symbol                         | value                     |
| $n_{ep}$           | 5000  | $\lambda_0$                 | 1                  | $\lambda_0$                    | 1                         |
| $n_f$              | 100 [42,43]   | $\lambda_1$                 | 200                | $\lambda_1$                    | 200                       |
| $\alpha$           | $1 \times 10^{-3}$  | $\lambda_2$                 | 100                | $\lambda_2$                    | 100                       |
| $n_{ba}$           | 64  | $\lambda_3$                 | $5 \times 10^{-3}$ | $\lambda_3$                    | $1 \times 10^{-4}$        |
| $T_{max}$          | 4 [42,43]   | $\lambda_4$                 | 0                  | $\lambda_{4,H}, \lambda_{4,L}$ | 10, 5                     |
| $\gamma$           | 1   | $\lambda_R$                 | 0                  | $\lambda_R$                    | 100%                      |
| $\varepsilon$      | $\begin{cases} 1 - \frac{e \times 90\%}{2000} & e < 2000 \\ 10\% & e \geq 2000 \end{cases}$ | $R$                         | 1                  | $R$                            | $\frac{ N_a }{\max( N )}$ |

parameter  $\lambda_{4,H}$ , and the low-cost parameter  $\lambda_{4,L}$ , respectively. In order to consider the higher costs of removing columns with higher internal force levels, the reduction factor  $R$  is taken as the ratio of  $|N_a|$  to  $\max(|N|)$  in Task 2, where  $N_a$  is the axial force of the column to be removed by action  $a_t$  at the current step  $t$ , and  $N$  is the set of axial forces of elements in  $\Omega_r$ .

The training history of Tasks 1 and 2 are plotted in Figs. 4 and 5, respectively, where the value of the loss function is plotted by the logarithmic scale. Notably, the reward and loss function of a half-trained agent needs to be evaluated using the greedy policy to reflect the best behavior, i.e., the additional computational cost is introduced. Therefore, the reward and the loss function are evaluated every 10 episodes in order to accelerate the training process. The training of the agent takes about 14.8 and 37.9 h on a laptop computer with a CPU of Intel(R) Core(TM) i7-7700 @3.60 GHz and a GPU of NVIDIA GeForce GTX 1080 for Tasks 1 and 2, respectively. Both the CPU and GPU have participated in the computation. It can be observed in Figs. 4 and 5 that the reward gradually increases and converges at a high value after 2000 episodes; besides, the loss function is almost 0 after about 400 episodes.

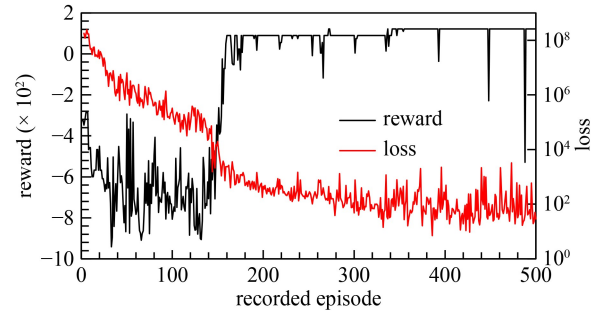
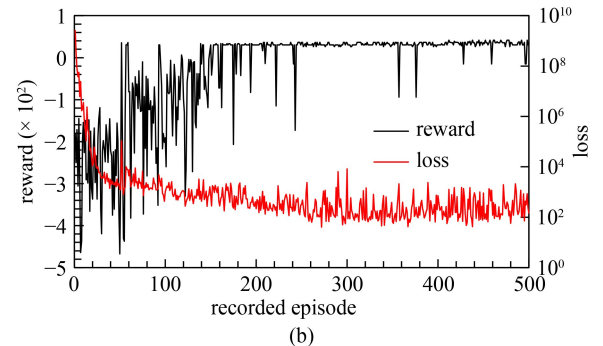
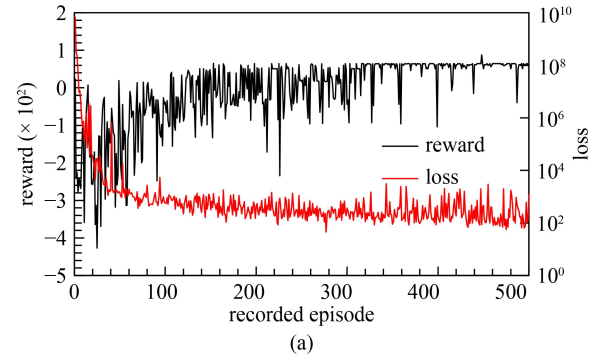


Fig. 4 Training history of Task 1.

Fig. 5 Training history of Task 2. (a) With high-cost parameter  $\lambda_{4,H}$ ; (b) with low-cost parameter  $\lambda_{4,L}$ .

### 4.3 Testing

Let  $\pi_1^*$ ,  $\pi_{2,H}^*$ , and  $\pi_{2,L}^*$  denote the policies with the highest recorded reward in Figs. 4, 5(a), and 5(b), respectively. This section tests their behavior on the  $4 \times 5$  frame used for training, a smaller-sized  $3 \times 4$  frame, and a larger-sized irregular frame shown in Fig. 6 without re-training.

#### a) Task 1—policy $\pi_1^*$

The states, including the estimated  $Q$  values provided by the trained agent with policy  $\pi_1^*$ , are plotted in Figs. 7–9 for the 3 different-sized frames. The greedy policy is adopted to remove the column with the highest  $Q$  value. In each step, the column to be removed is highlighted in red. For comparison, the states of a human-specified policy that exchanges the sequence of steps 1 and 2 of policy  $\pi_1^*$  are shown in Fig. 10. The reward parameters of the steps in Figs. 7 and 10 are tabulated in Table 3.

From Figs. 7–10 and Table 3, the following conclusions can be drawn:

1) By utilizing the GE technique, the agent is able to handle different-sized structures without re-training, as expected in Subsection 3.3. If GE is not employed, the size of the neural network parameters should be

associated with the number of removable elements  $n_e$ , and the trained agent cannot deal with structures with a different value of  $n_e$  since the matrix multiplication cannot be performed;

2) The agent provides a higher  $Q$  value for the ground-floor column on the right side of Fig. 9(a). As the internal force distribution of the irregular frame is significantly different from that of the  $4 \times 5$  symmetric frame used for training, it can be concluded that the agent has learned robust knowledge to adapt to both different-sized and irregular structures;

3) Although no information on symmetry has been introduced in the training process, the  $Q$  values provided by the trained agent are symmetric in symmetric structures, which indicates that the  $Q$  values are accurate for all elements. Besides, it can also be concluded that the PER method and the epsilon-greedy policy are effective in training a robust agent for the task.

4) The  $Q$  values will vary after the transition of the state. For example, the  $Q$  value of the column right above the removed column becomes smaller, as shown in Fig. 7(b). This is because removing this column in the next step may violate the anti-progressive collapse

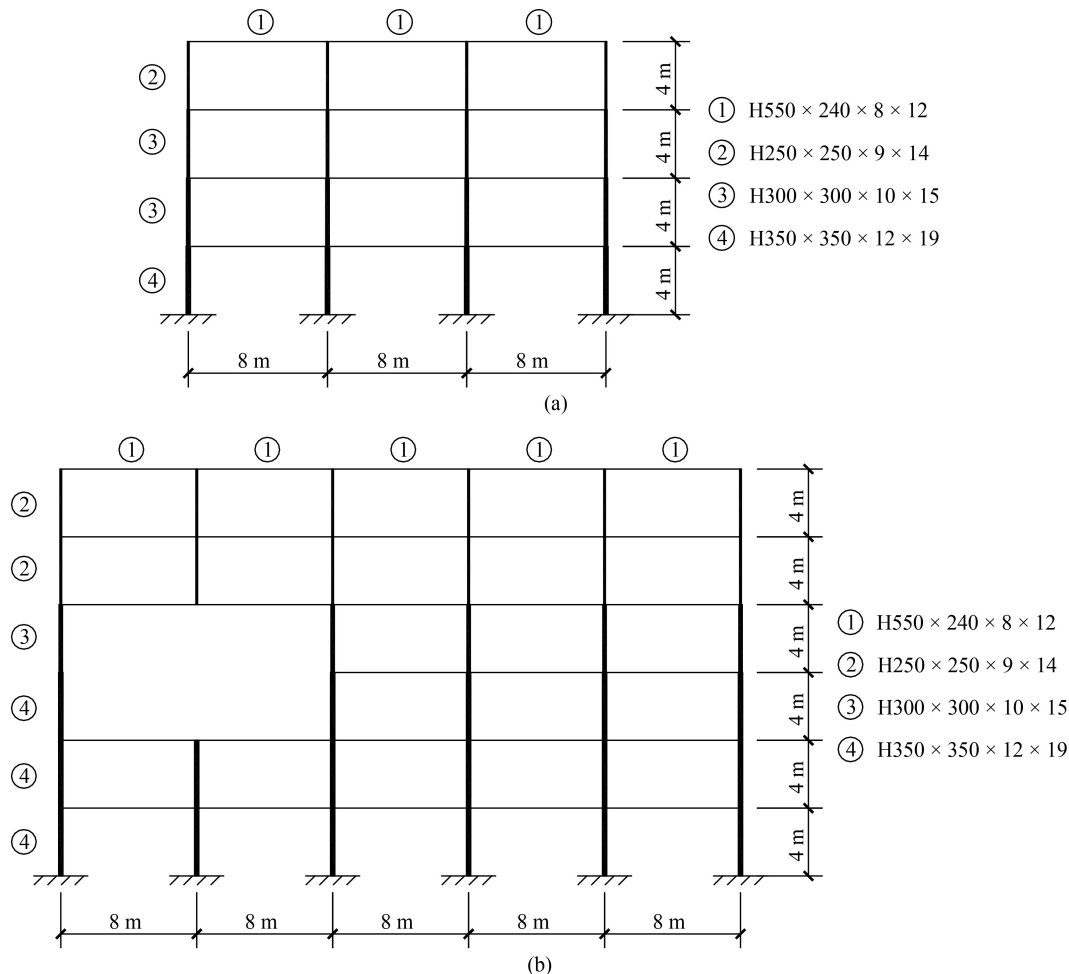
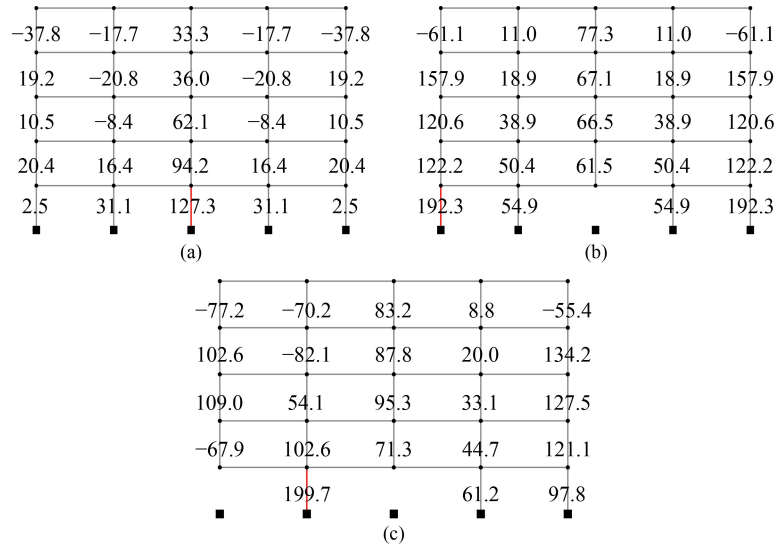
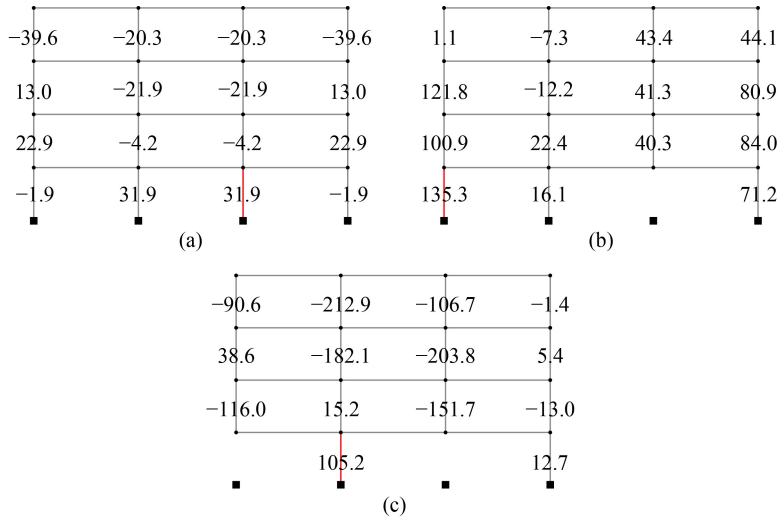


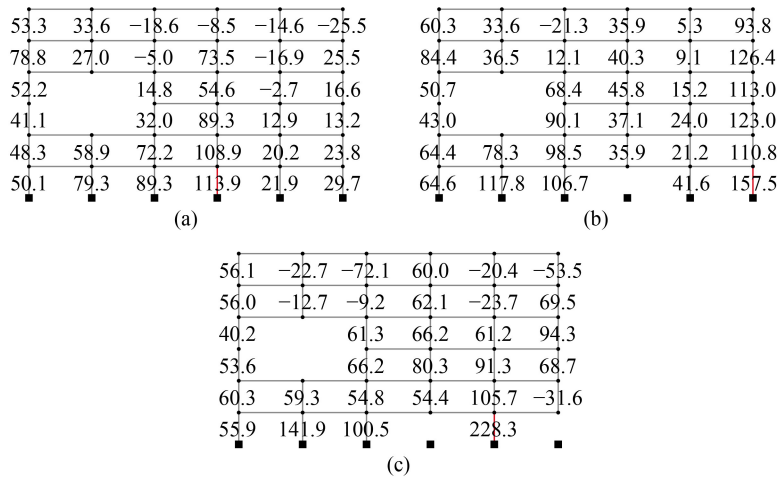
Fig. 6 Different-sized frames for testing. (a) Smaller-sized  $3 \times 4$  frame; (b) larger-sized irregular frame.



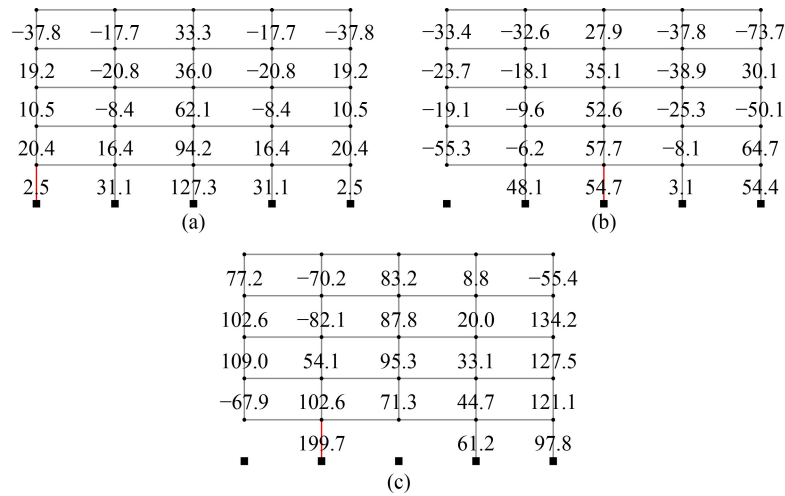
**Fig. 7** States when testing the agent with policy  $\pi_1^*$  on the  $4 \times 5$  frame. (a) Step 1; (b) Step 2; (c) Step 3.



**Fig. 8** States when testing the agent with policy  $\pi_1^*$  on the  $3 \times 4$  frame. (a) Step 1; (b) Step 2; (c) Step 3.



**Fig. 9** States when testing the agent with policy  $\pi_1^*$  on the irregular frame. (a) Step 1; (b) Step 2; (c) Step 3.



**Fig. 10** States when using a human-specified policy on the  $4 \times 5$  frame. (a) Step 1; (b) Step 2; (c) Step 3.

**Table 3** Comparison of reward parameters under different policies adopted in the  $4 \times 5$  frame for Task 1

| step | $k_{\text{sen},a}$ |                 | $r_{t+1}$   |                 | $R_{b,e}$ |                 |
|------|--------------------|-----------------|-------------|-----------------|-----------|-----------------|
|      | $\pi_1^*$          | human-specified | $\pi_1^*$   | human-specified | $\pi_1^*$ | human-specified |
| 1    | 69.63              | 67.07           | -70.60      | -71.51          | 0         | 0               |
| 2    | 545.64             | 25.90           | -6.53       | -87.85          | 0         | 0               |
| 3    | 942.35             | 942.35          | -0.90 + 200 | -0.90 + 200     | 100%      | 100%            |
| sum  | —                  | —               | 121.97      | 39.74           | —         | —               |

requirement by the adjacent beams; however, the majority of the remaining structure is intact, namely local collapse occurs, and the final positive reward is low because  $R_{b,e}$  is close to 0. Therefore, the trained agent is adaptive to the transition of different states, and the  $Q$  values can serve as a model-dependent index for the identification of critical elements;

5) By comparing policy  $\pi_1^*$  and the human-specified policy generated by exchanging the sequence of the first two actions, it can be seen that the final states are identical, and the final  $R_{b,e}$  of both policies equals 100%. Nonetheless, the sensitivity indices of actions taken by the agent with policy  $\pi_1^*$  are higher, resulting in a higher total reward. This is because the element removing sequence can significantly influence the internal force redistribution of intermediate states. As we expect a higher severity of the internal force redistribution for a specific state when identifying critical elements, it is reasonable to conclude that policy  $\pi_1^*$  is superior to the human-specified policy. In other words, the agent with policy  $\pi_1^*$  has successfully learned to predict the future consequences of the removal of an element; thus, the critical elements can be identified in sequence by the trained agent.

Notably, the structural analysis is conducted only once for evaluating the  $Q$  values of all columns. Therefore, the computational cost has been significantly reduced since evaluating the sensitivity index for all columns requires

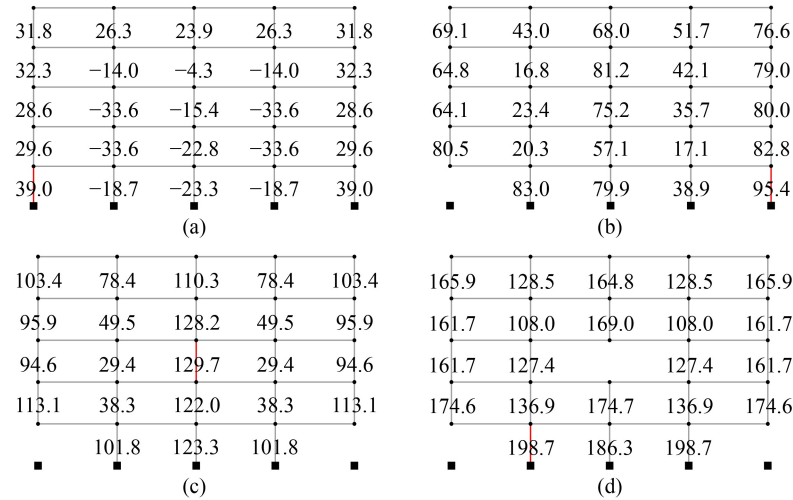
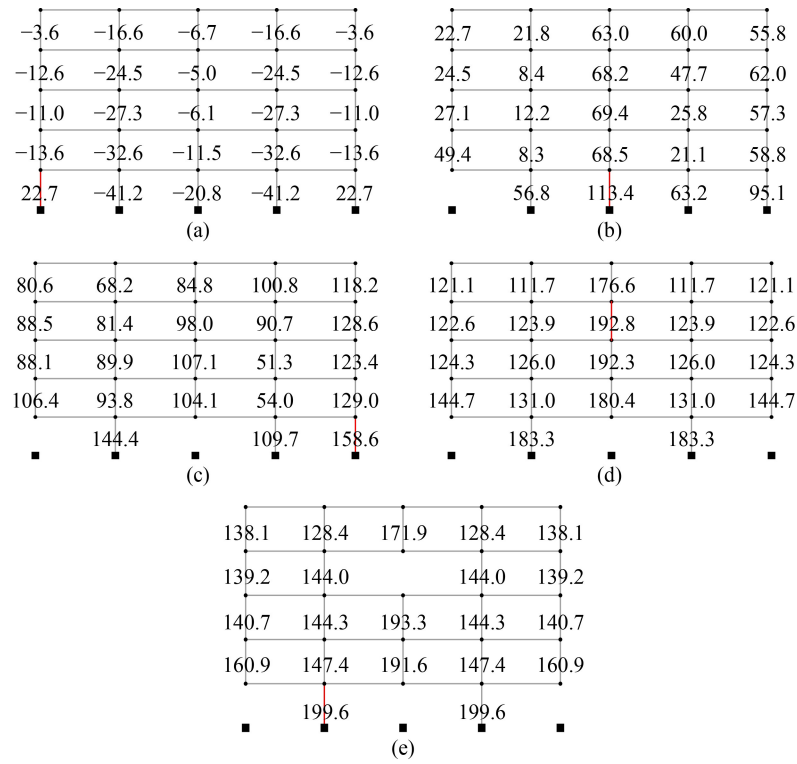
$n_e$  times of structural analysis. A more specific comparison of computational efficiency is given in Subsection 4.4.

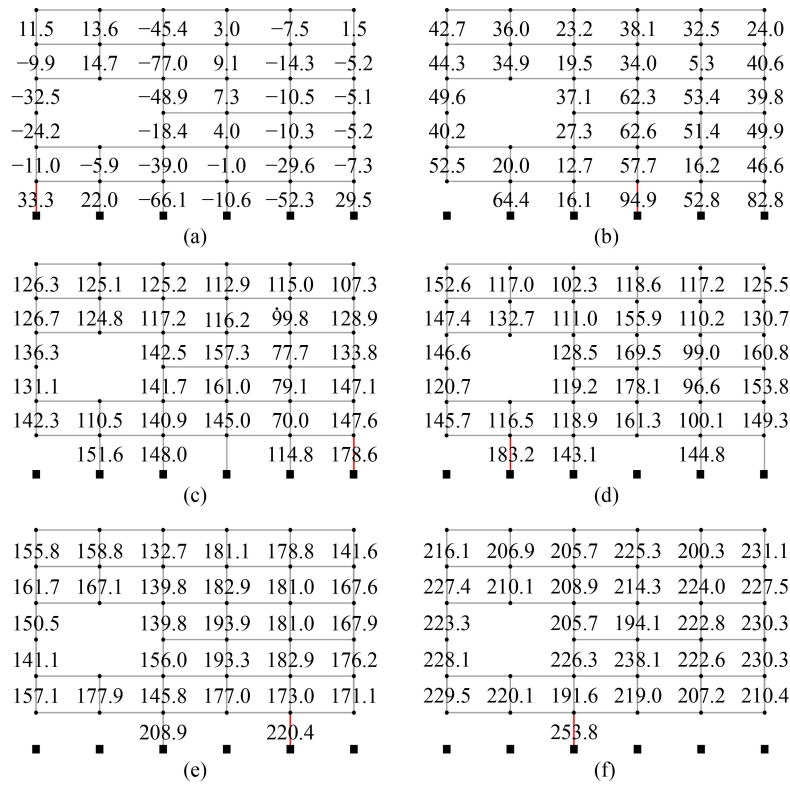
b) Task 2— $\pi_{2,H}^*$  and  $\pi_{2,L}^*$

The states, including the estimated  $Q$  values provided by the trained agent with policies  $\pi_{2,H}^*$  and  $\pi_{2,L}^*$  are plotted in Figs. 11 and 12 for the  $4 \times 5$  frame, respectively. As policies  $\pi_1^*$  and  $\pi_{2,H}^*$  both lead to a final  $R_{b,e}$  of 100%, Table 4 tabulates  $k_{\text{sen},a}$ ,  $r_{t+1}$ , and  $R$  under policies  $\pi_1^*$  and  $\pi_{2,H}^*$  for comparison. Note that reward  $r_{t+1}$  is calculated based on the high-cost hyperparameters of Task 2. Figures 13 and 14 show the behavior of the agent with policies  $\pi_{2,H}^*$  and  $\pi_{2,L}^*$  on the irregular frame without re-training. Note that only the sequence of removed columns is given in Fig. 14 since the number of actions is large.

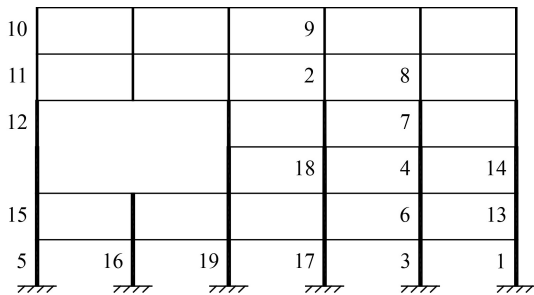
From Figs. 11–14 and Table 4, the following conclusions can be drawn.

1) By introducing a non-zero constant cost  $\lambda_4$  and a non-zero reduction factor concerning the internal force level  $R$ , the trained agent for Task 2 behaves differently compared to that for Task 1, and policy  $\pi_{2,H}^*$  is superior to  $\pi_1^*$  for Task 2 with respect to the total reward. Specifically, the agent with policy  $\pi_1^*$  only removes ground-floor columns since their sensitivity indices are quite high. However, since the reduction factor  $R$  is included in Task 2, the agent with policies  $\pi_{2,H}^*$  and  $\pi_{2,L}^*$  tend to remove the columns with a lower internal force level, e.g., Steps 1–3 in Fig. 11, or remove an upper-floor

**Table 4** Comparison of  $k_{\text{sen},a}$  and  $r_{t+1}$  under different policies adopted in the  $4 \times 5$  frame for Task 2



**Fig. 13** Behavior of agent with policy  $\pi_{2,H}^*$  for Task 2 on the irregular frame. (a) Step 1; (b) Step 2; (c) Step 3; (d) Step 4; (e) Step 5; (f) Step 6.



**Fig. 14** Behavior of agent with policy  $\pi_{2,L}^*$  for Task 2 on the irregular frame.

column in order to increase the sensitivity index of the next action, e.g., Step 4 in Fig. 11;

2) The symmetric  $Q$  values in symmetric states shown in Figs. 11(a), 11(d), and 11(e) also indicate that the training process is robust and the  $Q$  values for all actions are accurate;

3) The trained agents can produce reasonable results for different-sized and irregular structures without re-training, as shown in Figs. 13 and 14. As the internal force distribution of the irregular frame is significantly different from the symmetric  $4 \times 5$  frame used for training, it can be concluded that the agents have also been trained robustly;

4) Agents with different behaviors, i.e., agents with policies  $\pi_{2,H}^*$  and  $\pi_{2,L}^*$ , can be trained by adjusting

hyperparameter  $\lambda_4$ . Although the number of actions in the  $4 \times 5$  frame is similar for both agents, as shown in Figs. 11 and 12, the difference in behavior is more evident in the irregular frame, i.e., Figs. 13 and 14. While agent  $\pi_{2,H}^*$  only demolishes the ground-floor columns, agent  $\pi_{2,L}^*$  tends to demolish more upper-floor columns to reduce the axial force level of the ground-floor columns with a low sensitivity index at the initial stage; besides, it also removes the columns on both sides in order to increase the sensitivity index of the ground-floor columns to be removed. Hence, the different importance of safety and cost in Task 2 can be reasonably considered by varying hyperparameter  $\lambda_4$ .

We need to emphasize again that the behavior of the agent will be sensitive to the hyperparameters in the reward function. Nonetheless, according to the favorable performance of the trained agent which can be applied to different-sized structures without re-training, the hyperparameters of the reward function tabulated in Table 2 are recommended for studies with the same aim described in Section 2.

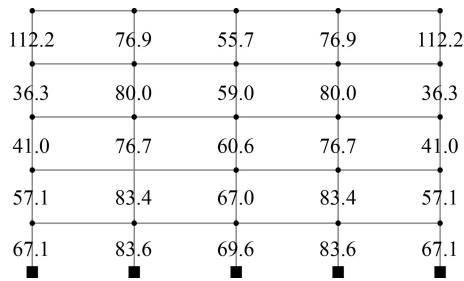
#### 4.4 Comparison of computational efficiency and decisions

In order to illustrate the advantage of the proposed method, this section compares the computational efficiency of the proposed deep RL-based method and the

existing sensitivity index-based ALP method, i.e., the indices are calculated based on Eq. (27).

For the  $4 \times 5$  planar frame in Fig. 3, Fig. 15 shows the sensitivity indices of the columns when all the columns are traversed. The data in Fig. 15 require 26 ( $n_e = 25$ ) non-linear analyses, and takes about 26.61 s on the same laptop computer used for training the RL agent. Meanwhile, the data shown in Figs. 7(a), 11(a) or 12(a) only require single non-linear analysis, which takes only about 1.37 s with the trained agent. Table 5 tabulates the computational time of the  $Q$  values calculated by the proposed method and the sensitivity index calculated by the conventional method, i.e., Eq. (27), for the original state of the 3 numerical examples. By comparing the increase rate of computational efficiency  $IR$  and  $n_e$ , it can be concluded that the computational efficiency of the proposed method is increased by approximately  $n_e \times 100\%$  with respect to the conventional method. In Table 5, the offline training time of the RL agent is also given. However, we need to note again that the trained agent can be applied to different-sized problems without re-training.

Besides, as criticized by Jiang et al. [20] that the indices shown in Fig. 15 are short-sighted and cannot be directly used for decision-making for determining the most expected collapse scenario of CEI and DP. For example, although the corner columns at the top floor have the highest index of 112.2, a local collapse will occur if either of them is removed, and only 1 of the 20 beams will exceed the rotational limit specified by Eq. (23). On the other hand, by tuning the hyperparameters in the reward function, the proposed deep RL-based method can train agents with different behaviors, i.e., agents producing different  $Q$  values, for different tasks, including CEI and DP. Note that the  $Q$ -values have taken the ultimate



**Fig. 15** Sensitivity index evaluated by the conventional ALP method.

**Table 5** Computational time of the numerical examples

| numerical example  | $n_e$ | training time (h) | proposed method (s) | conventional method (s) | $IR/100\%$ |
|--------------------|-------|-------------------|---------------------|-------------------------|------------|
| $4 \times 5$ frame | 25    | 14.8              | 1.37                | 26.61                   | 18.42      |
| $3 \times 4$ frame | 16    |                   | 1.07                | 15.13                   | 13.14      |
| irregular frame    | 34    |                   | 1.63                | 76.65                   | 46.02      |

collapse scenario into consideration, as discussed in Subsection 3.5 and validated in Subsection 4.3.

## 5 Conclusions

This paper proposes a framework for CEI and DP of frame structures. Innovative quantitative indices for elements characterizing their importance with respect to the ultimate collapse scenario are proposed using RL and GE. Through numerical examples, the following conclusions are obtained.

1) In the training process of the numerical examples, the agent can converge at a high total reward and a loss function close to zero, indicating that the formulation of the RL task is feasible.

2) The trained agent can also handle environments with different-sized action spaces, i.e., structures with different number of elements, owing to the utilization of the GE technique.

3) The PER method and the epsilon-greedy policy are proved to be effective in training robust agents.

4) By adequately setting the hyperparameters in the reward function, the  $Q$  values provided by the trained agent can serve as quantitative indices for CEI and DP of frame structures. For both tasks, the  $Q$  values have considered the impact of the ultimate collapse scenario, the sensitivity index of the removed element, and the sequence of removed elements. For DP of frames, the importance of the severity of collapse can be increased by adjusting the hyperparameter  $\lambda_R$  in the reward function in order to ensure an overall collapse. Besides, different human-defined importance of safety and cost in the task of DP can be incorporated by adjusting the hyperparameter  $\lambda_4$  in the reward function.

5) The computational efficiency of the proposed deep RL-based method is significantly increased by about  $n_e \times 100\%$  compared with the conventional sensitivity index-based method. Besides, the proposed indices, i.e., the  $Q$  values obtained by the RL agent, are shown to be superior to existing short-sighted indices and can be directly used for decision-making in the tasks.

**Acknowledgments** The authors gratefully acknowledge the financial support provided by the China Scholarship Council (CSC) during a visit of Shaojun Zhu to Kyoto University (No. 201906260152). The second author acknowledges the support of JSPS KAKENHI (Grant No. JP20H04467). The third author acknowledges the support of Grant-in-Aid for Young Scientists (Start-up) (Grant No. JP21K20461).

## Notations

$n_e$ : total number of removable elements  
 $\xi_0$ : global response of the original structure with the  $i$ th element removed

$\xi_i$ : global response of the remaining structure with the  $i$ th element removed  
 $\xi_{j,0}$ : response of the  $j$ th element without any element removed  
 $\xi_{j,i}$ : response of the  $j$ th element with the  $i$ th element removed  
 $\Omega_r$ : set of indices of the remaining elements that can be removed  
 $\gamma_{i,t}$ : sensitivity index of the  $i$ th element at the  $t$ th element removal step  
 $\xi_{0,t}^{(i)}$ : structural global response at the  $t$ th element removal step before the loss of the  $i$ th element  
 $\xi_{i,t}$ : structural global response at the  $t$ th element removal step after the loss of the  $i$ th element  
 $\xi_{j,0,t}^{(i)}$ : response of the  $j$ th element at the  $t$ th element removal step before the loss of the  $i$ th element  
 $\xi_{j,i,t}$ : response of the  $j$ th element at the  $t$ th element removal step after the loss of the  $i$ th element  
 $s_t$ : current state of the environment  
 $a_t$ : action at the  $t$ th step  
 $\pi$ : policy  
 $r_t$ : reward at the  $t$ th step  
 $\mathcal{A}$ : action space  
 $\mathbf{v}_i$ : feature vector of the  $i$ th joint  
 $n_n$ : number of joints in the structure  
 $\bar{x}_i$ : normalized horizontal coordinate of the  $i$ th joint in the  $x$  direction  
 $\bar{y}_i$ : normalized vertical coordinate of the  $i$ th joint in the  $y$  direction  
 $x_i$ : actual coordinate of the  $i$ th joint in the  $x$  direction  
 $y_i$ : actual coordinate of the  $i$ th joint in the  $y$  direction  
 $\mathbf{x}$ : set of horizontal joint coordinates  
 $\mathbf{y}$ : set of vertical joint coordinates  
 $\Delta x$ : range of the horizontal joint coordinates  
 $\Delta y$ : range of the vertical joint coordinates  
 $\bar{d}_i$ : normalized distance between the  $i$ th joint and the nearest support  
 $F_i$ : downward concentrated load at the  $i$ th joint  
 $\mathbf{m}_j$ : feature vector of the  $j$ th element  
 $\text{Sel}_j$ : existence/nonexistence index of the  $j$ th element  
 $l_j$ : length of the  $j$ th element  
 $A_j$ : area of the  $j$ th element  
 $I_{z,j}$ : moment of inertia about the strong axis of the  $j$ th element  
 $f_{y,j}$ : yield strength of the material of the  $j$ th element  
 $C_j$ : strain energy of the  $j$ th element  
 $R_{C,j}$ : strain energy ratio of the  $j$ th element  
 $C_{\max}$ : maximum strain energy of a single element out of the  $n_e$  elements  
 $q_j$ : intensity of the downward distributed load per unit length of the  $j$ th element  
 $\boldsymbol{\mu}_j$ : comprehensive feature vector of the  $j$ th element  
 $n_f$ : size of the comprehensive feature vector  
 $T_{\max}$ : maximum number of iterations in the GE process  
 $\boldsymbol{\mu}_j^{(T)}$ : comprehensive feature vector of the  $j$ th element at

the  $t$ th iteration  
 $\mathbf{h}_1, \mathbf{h}_2, \mathbf{h}_3^{(T)}, \mathbf{h}_4^{(T)}$ : intermediate vectors  
 $\mathbf{v}_{j,i}$ : joint feature vector of the  $i$ th end of the  $j$ th element  
 $\Phi_{j,i}$ : set of element indices connected to the  $i$ th end of the  $j$ th element except for the  $j$ th element itself  
 $\boldsymbol{\theta}_1\text{--}\boldsymbol{\theta}_6$ : weight matrices of the fully-connected neural network layers in the GE network for the comprehensive feature vector  
 $n_v$ : size of the joint feature vectors  
 $n_m$ : size of the element feature vectors  
 $\hat{\boldsymbol{\mu}}^{(T)}$ : comprehensive feature matrix at the  $t$ th iteration  
 $Q_j$ : quantitative index for each action  
 $\tilde{\pi}(\cdot)$ : output of the DQN  
 $\boldsymbol{\theta}_7\text{--}\boldsymbol{\theta}_9$ : weight matrices of the neural network layers in the DQN  
 $\phi_b$ : plastic rotation of beam element  
 $\phi_{\text{lim}}$ : upper limit of the plastic rotation of beam element  
 $h$ : height of the beam cross-section  
 $\lambda_i$ : hyperparameters in the reward function  
 $n_{b,0}$ : number of isolated beams  
 $n_b$ : total number of beams  
 $R$ : reduction factor of the cost of removing an element  
 $k_{\text{sen},a}$ : strain energy sensitivity index of the action in the current step  
 $C_{j,0}$ : strain energy of the  $j$ th element before execution of the action  
 $C_{j,a}$ : strain energy of the  $j$ th element after execution of the action  
 $n_{b,e}$ : number of beams exceeding the plastic rotation upper limit  
 $\lambda_R$ : a hyperparameter to terminate the episode  
 $\boldsymbol{\Theta}$ : set of network parameters  
 $\hat{\boldsymbol{\Theta}}$ : the most recent set of network parameters  
 $L(\boldsymbol{\Theta})$ : loss function describing the difference between the  $q$  value and the actual reward  
 $\delta$ : temporal-difference error  
 $D$ : set of records  
 $n_D$ : maximum number of records  
 $n_{ba}$ : size of the batch  
 $p$ : possibility of adopting the policy  
 $\varepsilon$ : exploration rate  
 $n_{\text{ep}}$ : total number of episodes  
 $e$ : episode number  
 $N_a$ : axial force of the column to be removed by action  $a_i$  at the  $t$ th step  
 $N$ : set of axial forces of elements in the remaining columns  
 RL: reinforcement learning  
 GE: graph embedding  
 ALP: alternate load path  
 DP: demolition planning  
 CEI: critical element identification  
 MDP: Markov decision process  
 SL: supervised learning  
 DQN: deep Q network

## References

- Starossek U. *Progressive Collapse of Structures*. London: Thomas Telford, 2009
- Shi Y, Li Z, Hao H. A new method for progressive collapse analysis of RC frames under blast loading. *Engineering Structures*, 2010, 32(6): 1691–1703
- Zhao X, Yan S, Chen Y, Xu Z, Lu Y. Experimental study on progressive collapse-resistant behavior of planar trusses. *Engineering Structures*, 2017, 135: 104–116
- Izzuddin B A, Vlassis A G, Elghazouli A Y, Nethercot D A. Progressive collapse of multi-storey buildings due to sudden column loss—Part I: Simplified assessment framework. *Engineering Structures*, 2008, 30(5): 1308–1318
- Vlassis A G, Izzuddin B A, Elghazouli A Y, Nethercot D A. Progressive collapse of multi-storey buildings due to sudden column loss—Part II: Application. *Engineering Structures*, 2008, 30(5): 1424–1438
- Kiakoouri F, De Biagi V, Chiaia B, Sheidaii M R. Progressive collapse of framed building structures: Current knowledge and future prospects. *Engineering Structures*, 2020, 206: 110061
- Frangopol D M, Curley J P. Effects of damage and redundancy on structural reliability. *Journal of Structural Engineering*, 1987, 113(7): 1533–1549
- Jiang X, Chen Y. Progressive collapse analysis and safety assessment method for steel truss roof. *Journal of Performance of Constructed Facilities*, 2012, 26(3): 230–240
- Frangopol D M, Curley J P. Effects of damage and redundancy on structural reliability. *Journal of Structural Engineering*, 1987, 113(7): 1533–1549
- Fallon C T, Quil S E, Naito C J. Uniform pushdown approach for quantifying building-frame robustness and the consequence of disproportionate collapse. *Journal of Performance of Constructed Facilities*, 2016, 30(6): 04016060
- Biondini F, Frangopol D M, Restelli S. On structural robustness, redundancy and static indeterminacy. In: *Proceedings of the Structures Congress 2008*. Vancouver: American Society of Civil Engineers, 2008
- Starossek U, Haberland M. Approaches to measures of structural robustness. *Structure and Infrastructure Engineering*, 2011, 7(7–8): 625–631
- Bao Y, Main J A, Noh S Y. Evaluation of structural robustness against column loss: Methodology and application to RC frame buildings. *Journal of Structural Engineering*, 2017, 143(8): 04017066
- Chen C, Zhu Y, Yao Y, Huang Y. Progressive collapse analysis of steel frame structure based on the energy principle. *Steel and Composite Structures*, 2016, 21(3): 553–571
- Bazant Z P, Verdure M. Mechanics of progressive collapse: Learning from World Trade Center and building demolitions. *Journal of Engineering Mechanics*, 2007, 133(3): 308–319
- Luccioni B M, Ambrosini R D, Danesi R F. Analysis of building collapse under blast loads. *Engineering Structures*, 2004, 26(1): 63–71
- Sun J, Jia Y, Yao Y, Xie X. Experimental investigation of stress transients of blasted RC columns in the blasting demolition of buildings. *Engineering Structures*, 2020, 210: 110417
- Zhu J, Zheng W, Sneed L H, Xu C, Sun Y. Green demolition of reinforced concrete structures: Review of research findings. *Global Journal of Research in Engineering*, 2019, 19(4): 1–18
- Isobe D. An analysis code and a planning tool based on a key element index for controlled explosive demolition. *International Journal of High-Rise Buildings*, 2014, 3(4): 243–254
- Jiang J, Lu D, Lu X, Li G, Ye J. Research progress and development trends on progressive collapse resistance of building structures. *Journal of Building Structures*, 2022, 43(1): 1–28 (in Chinese)
- Ohsaki M. Random search method based on exact reanalysis for topology optimization of trusses with discrete cross-sectional areas. *Computers & Structures*, 2001, 79(6): 673–679
- Makode P V, Ramirez M R, Corotis R B. Reanalysis of rigid frame structures by the virtual distortion method. *Structural Optimization*, 1996, 11(2): 71–79
- Ellingwood B R, Dusenberry D O. Building design for abnormal loads and progressive collapse. *Computer-Aided Civil and Infrastructure Engineering*, 2005, 20(3): 194–205
- Chen C, Zhu Y, Yao Y, Huang Y, Long X. An evaluation method to predict progressive collapse resistance of steel frame structures. *Journal of Constructional Steel Research*, 2016, 122: 238–250
- Jiang L, Ye J. Risk-based robustness assessment of steel frame structures to unforeseen events. *Civil Engineering and Environmental Systems*, 2018, 35(1–4): 117–138
- Baker J W, Schubert M, Faber M H. On the assessment of robustness. *Structural Safety*, 2008, 30(3): 253–267
- Liao K W, Wen Y K, Foutch D A. Evaluation of 3D steel moment frames under earthquake excitations. II: Reliability and redundancy. *Journal of Structural Engineering*, 2007, 133(3): 471–480
- Feng D, Xie S, Xu J, Qian K. Robustness quantification of reinforced concrete structures subjected to progressive collapse via the probability density evolution method. *Engineering Structures*, 2020, 202: 109877
- Sutton R S, Barto A G. *Reinforcement Learning: An Introduction*. Cambridge: MIT press, 2018
- Jordan M I, Mitchell T M. *Machine learning: Trends, perspectives, and prospects*. Science, 2015, 349(6245): 255–260
- Sun H, Burton H V, Huang H. Machine learning applications for building structural design and performance assessment: State-of-the-art review. *Journal of Building Engineering*, 2021, 33: 101816
- Salehi H, Burgueño R. Emerging artificial intelligence methods in structural engineering. *Engineering Structures*, 2018, 171: 170–189
- Zhu S, Ohsaki M, Guo X. Prediction of non-linear buckling load of imperfect reticulated shell using modified consistent imperfection and machine learning. *Engineering Structures*, 2021, 226: 111374
- Xue J, Xiang Z, Ou G. Predicting single freestanding transmission tower time history response during complex wind input through a convolutional neural network based surrogate model. *Engineering Structures*, 2021, 233: 111859
- Guo H, Zhuang X, Rabczuk T. A deep collocation method for the bending analysis of Kirchhoff plate. *Computers, Materials & Continua*, 2019, 59(2): 433–456

36. Anitescu C, Atroshchenko E, Alajlan N, Rabczuk T. Artificial neural network methods for the solution of second order boundary value problems. *Computers, Materials & Continua*, 2019, 59(1): 345–359
37. Samaniego E, Anitescu C, Goswami S, Nguyen-Thanh V M, Guo H, Hamdia K, Zhuang X, Rabczuk T. An energy approach to the solution of partial differential equations in computational mechanics via machine learning: Concepts, implementation and applications. *Computer Methods in Applied Mechanics and Engineering*, 2020, 362: 112790
38. Zhuang X, Guo H, Alajlan N, Zhu H, Rabczuk T. Deep autoencoder based energy method for the bending, vibration, and buckling analysis of Kirchhoff plates with transfer learning. *European Journal of Mechanics. A, Solids*, 2021, 87: 104225
39. Guo H, Zhuang X, Chen P, Alajlan N, Rabczuk T. Stochastic deep collocation method based on neural architecture search and transfer learning for heterogeneous porous media. *Engineering with Computers*, 2022, 1–26
40. Guo H, Zhuang X, Chen P, Alajlan N, Rabczuk T. Analysis of three-dimensional potential problems in non-homogeneous media with physics-informed deep collocation method using material transfer learning and sensitivity analysis. *Engineering with Computers*, 2022, 1–22
41. Li S, Snaiki R, Wu T. A knowledge-enhanced deep reinforcement learning-based shape optimizer for aerodynamic mitigation of wind-sensitive structures. *Computer-Aided Civil and Infrastructure Engineering*, 2021, 36(6): 733–746
42. Hayashi K, Ohsaki M. Reinforcement learning and graph embedding for binary truss topology optimization under stress and displacement constraints. *Frontiers in Built Environment*, 2020, 6: 59
43. Zhu S, Ohsaki M, Hayashi K, Guo X. Machine-specified ground structures for topology optimization of binary trusses using graph embedding policy network. *Advances in Engineering Software*, 2021, 159: 103032
44. Makarov I, Kiselev D, Nikitinsky N, Subelj L. Survey on graph embeddings and their applications to machine learning problems on graphs. *PeerJ. Computer Science*, 2021, 7: e357
45. Dai H, Khalil E B, Zhang Y, Dilkina B, Song L. Learning combinatorial optimization algorithms over graphs. 2017, arXiv: 1704.01665
46. Nguyen T T, Reddi V J. Deep reinforcement learning for cyber security. 2019, arXiv: 1906.05799
47. Fan C, Zeng L, Sun Y, Liu Y. Finding key players in complex networks through deep reinforcement learning. *Nature Machine Intelligence*, 2020, 2(6): 317–324
48. Nafday A M. System safety performance metrics for skeletal structures. *Journal of Structural Engineering*, 2008, 134(3): 499–504
49. Kiakojouri F, Sheidaii M R, De Biagi V, Chiaia B. Progressive collapse of structures: A discussion on annotated nomenclature. *Structures*, 2021, 29: 1417–1423
50. Watkins C J C H, Dayan P. Q-learning. *Machine Learning*, 1992, 8(3–4): 279–292
51. Williams R J. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 1992, 8(3–4): 229–256
52. Schulman J, Wolski F, Dhariwal P, Radford A, Klimov O. Proximal policy optimization algorithms. 2017, arXiv: 1707.06347
53. UFC 4-023-03 (Change 3). *Design of Buildings to Resist Progressive Collapse*. Washington, D.C.: United States Department of Defense, 2016
54. CECS 392:2014. *Code for Anti-collapse Design of Building Structures*. Beijing: China Planning Press, 2014 (in Chinese)
55. Mnih V, Kavukcuoglu K, Silver D, Rusu A A, Veness J, Bellemare M G, Graves A, Riedmiller M, Fidjeland A K, Ostrovski G, Peterson S, Beattie C, Sadik A, Antonoglou I, King H, Kumaran D, Wierstra D, Legg S, Hassabis D. Human-level control through deep reinforcement learning. *Nature*, 2015, 518(7540): 529–533
56. Rumelhart D E, Hinton G E, and Williams R J. Learning representations by back-propagating errors. *Nature*, 1986, 323(6088): 533–536
57. Tieleman T, Hinton G. Lecture 6. 5-RMSProp: Divide the gradient by a running average of its recent magnitude. COURSE: Neural Networks for Machine Learning, 2012, 4: 26–31
58. Schaul T, Quan J, Antonoglou I, Silver D. Prioritized experience replay. 2015, arXiv: 1511.05952
59. GSA. *Progressive Collapse Analysis and Design Guidelines for New Federal Office Buildings and Major Modernization Projects*. USA: US General Services Administration (GSA), 2005
60. ANSYS® Multiphysics 19.0. Canonsburg, PA: Ansys Inc., 2018
61. Riks E. An incremental approach to the solution of snapping and buckling problems. *International Journal of Solids and Structures*, 1979, 15(7): 529–551
62. Crisfield M A. An arc-length method including line searches and accelerations. *International Journal for Numerical Methods in Engineering*, 1983, 19(9): 1269–1289
63. Kaszynski A. PyMAPDL (version 0.59.4). Zenodo, 2021