Minimum-volume design of steel frames using reinforcement learning

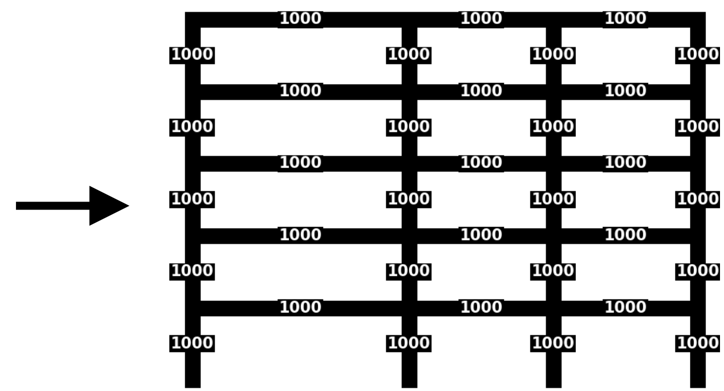
Kazuki Hayashi and Makoto Ohsaki

Department of Architecture and
Architectural Engineering, Kyoto University

Why structural optimization x RL?

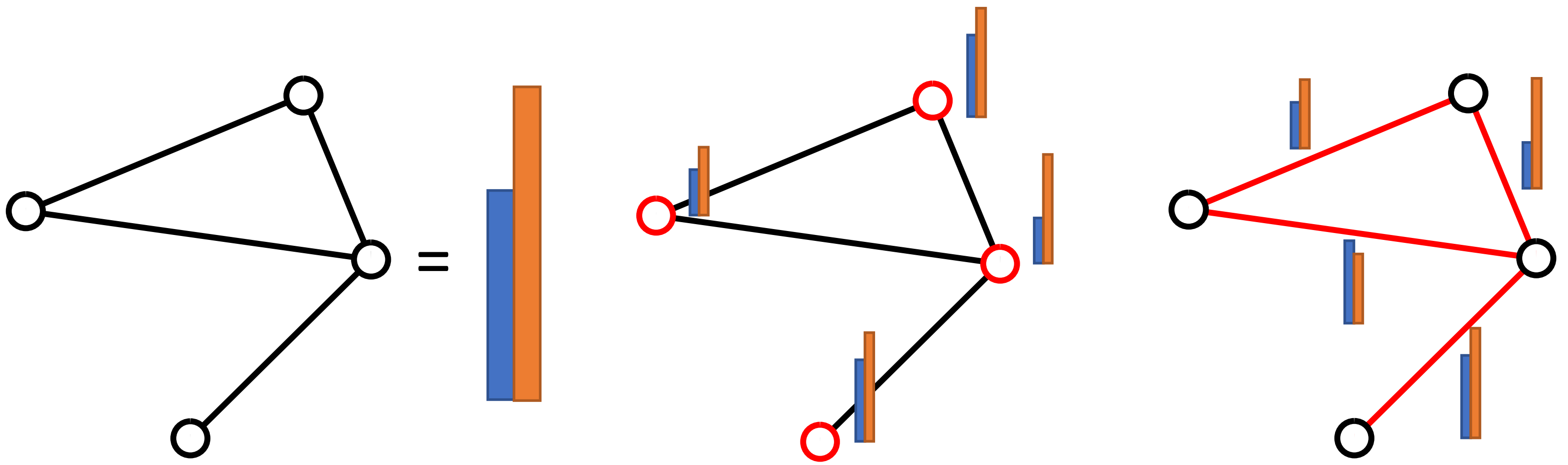
1. Capability of handling difficult problem
2. Inspiration from unexpected (and good) solutions
3. Reduction of computational cost

→ Simulate structural engineers' design process



Graph embedding (GE)

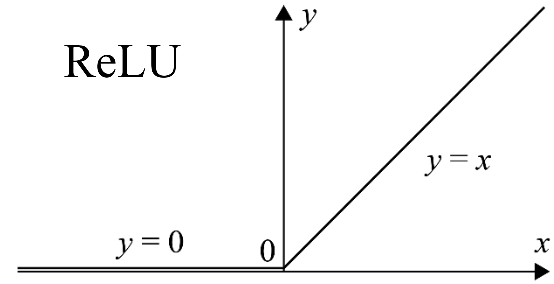
= Convolution for graphs to extract features



	Whole graph embedding	Node embedding	Edge embedding
application	Comparison of chemical structures	Travelling Salesman problem	Link prediction between nodes
methods	Graph2vec (Narayanan et al., 2017) UGRAPHEMB (Bai et al., 2019)	Structure2Vec (Hanjun et al., 2016) DeepWalk (Perozzi et al., 2014)	Edge2Vec (Wang et al., 2020) <u>edge input → edge feature</u>

Propose a new method for
<u>node and edge input → edge feature</u>

Graph embedding



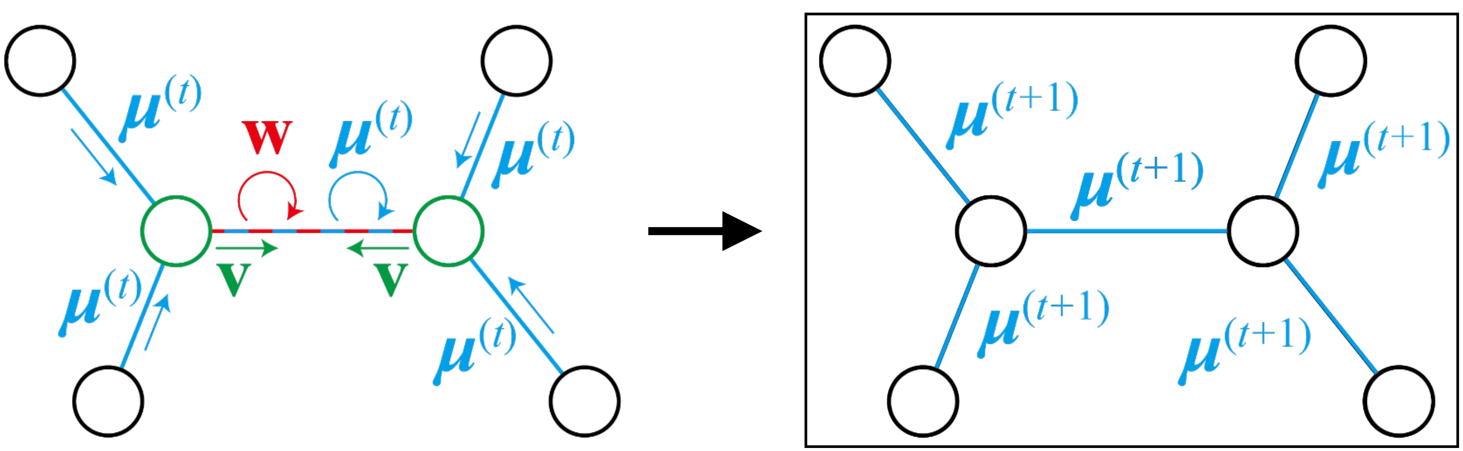
w_{i_m} : input of member i_m

$v_{i_\mathrm{m},j}$: input of j-th end of member i_m

μ_{i_m} : extracted feature of member i_m

Trainable parameters to be adjusted

$$\mu_{i_\mathrm{m}}^{(t+1)} \leftarrow \mathrm{ReLU}\left(\theta_1 w_{i_\mathrm{m}} + \theta_2 \sum_{j=1}^{2} \mathrm{ReLU}(\theta_3 v_{i_\mathrm{m},j}) + \theta_4 \mu_{i_\mathrm{m}}^{(t)} + \theta_5 \sum_{i=1}^{2} \mathrm{ReLU}\left(\theta_6 \sum_{k \in \aleph\left(v_{i_\mathrm{m},j}\right)} \mu_k^{(t)}\right)\right)$$



Repeat until obtaining $\mu^{(4)}$

Expression of action value Q using μ

- $Q(s,a)$: value to change design of member a at state s

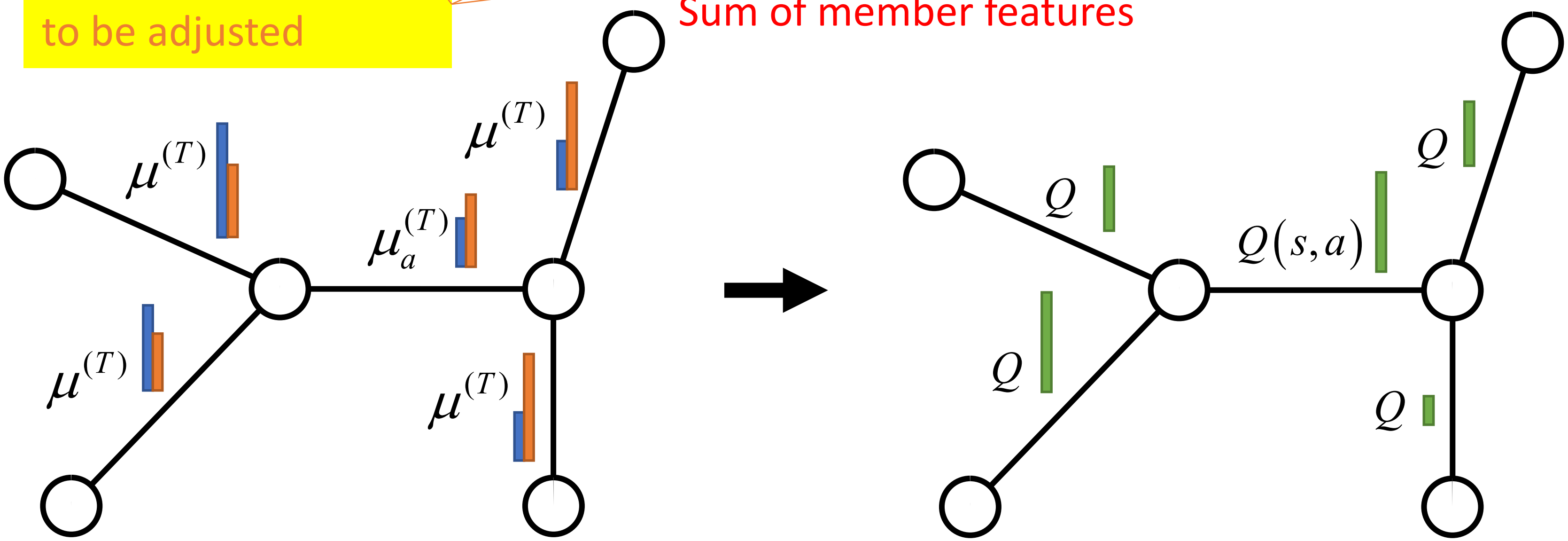
concatenation

$$Q(s,a) = \theta_7^{\mathrm{T}}\left[\mathrm{ReLU}\left(\theta_8 \sum_{k \in \mathbf{V}} \mu_k^{(T)}\right), \mathrm{ReLU}\left(\theta_9 \mu_a^{(T)}\right)\right]$$

Trainable parameters to be adjusted

Feature of member a

Sum of member features



Training of $\mathbf{\Theta} = \{\theta_1, \cdots, \theta_9\}$

Observed reward + γ Max. Q at the next state

Q at the current state-action pair

- Q-Learning (Watkins, 1989)

$$Q(s,a) \leftarrow Q(s,a) + \alpha\left(r(s') + \gamma \max_a Q(s',a) - Q(s,a)\right)$$

- Deep-Q Network (Mnih, 2015)

$$\text{minimize } L(\mathbf{\Theta}) = \left(r(s') + \gamma \max_{a \in \Omega_{s'}} Q\left(s',a \mid \tilde{\mathbf{\Theta}}\right) - Q(s,a \mid \mathbf{\Theta})\right)^2$$

discount rate $\in [0,1]$ previous trainable parameters

- The parameters $\mathbf{\Theta}$ are updated to minimize the loss using RMSprop (Tieleman and Hinton, 2012)

Tieleman, T. and Hinton, G.: Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude. COURSERA: Neural Networks for Machine Learning, Vol. 4, pp. 26-30, 2012.

Cross-section optimization of steel frames

$$\text{minimize } V(\mathbf{A})$$ Total structural volume

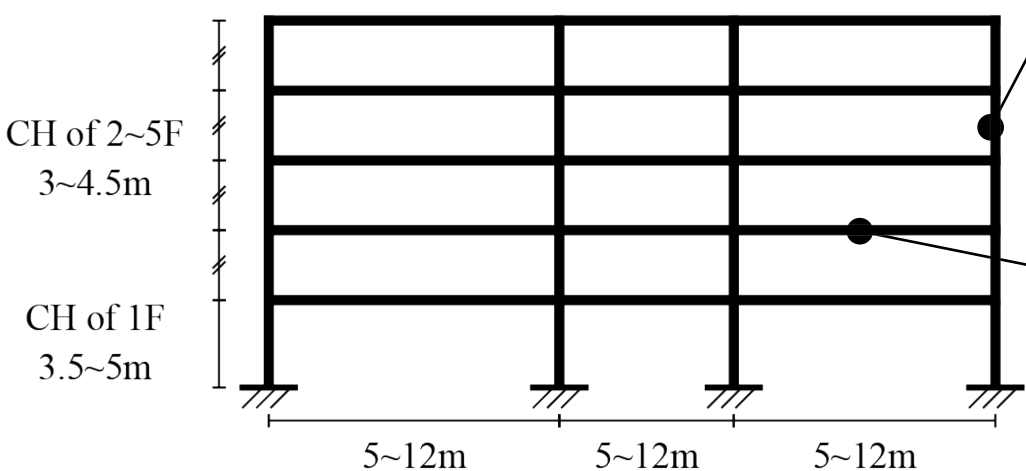
$$\text{subject to } \sigma_i(\mathbf{A}) \le \bar{\sigma}_i \quad (i \in \text{all members})$$ Member stress

$$u_i^{\mathrm{c}}(\mathbf{A})/L_i \le 1/200 \quad (i \in \text{columns})$$ Deformation of columns

$$v_i^{\mathrm{b}}(\mathbf{A})/L_i \le 1/300 \quad (i \in \text{beams})$$ Deformation of beams

$$\beta_j(\mathbf{A}) \ge 1.5 \quad (j \in \text{middle-layer nodes})$$ Column-to-beam overstrength factor

Consider both long-term and short-term loads

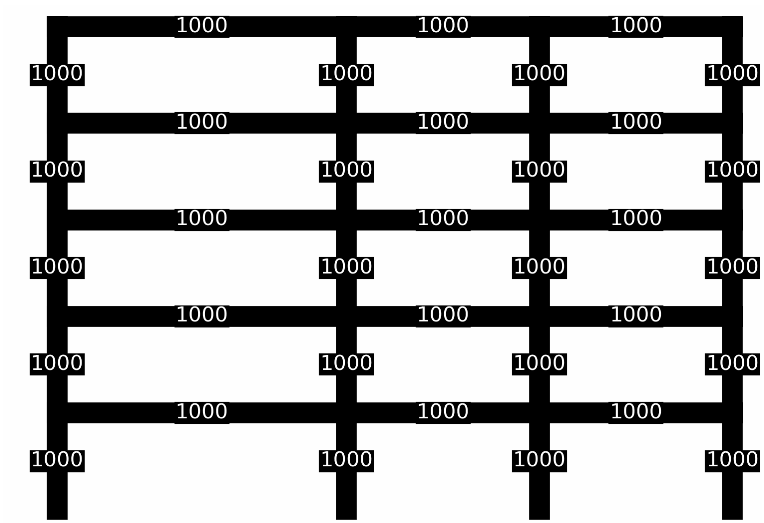


17 types of square pipes from 200 to 1000 mm at the interval of 50 mm

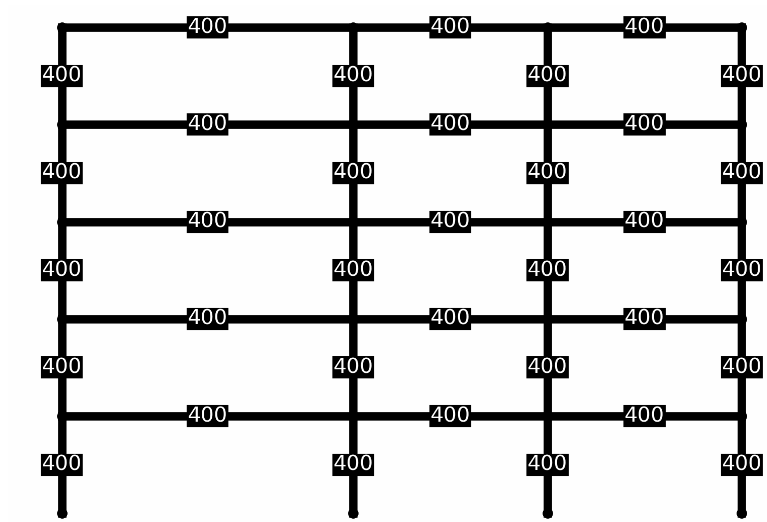
17 types of I-beams from 200 to 1000 mm at the interval of 50 mm

Two training cases

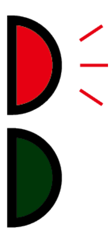
Reduce the size

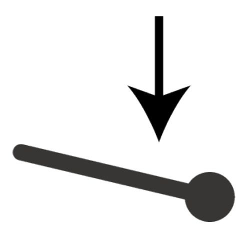


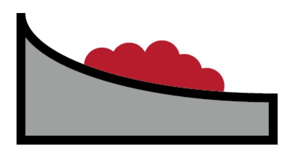
Increase the size



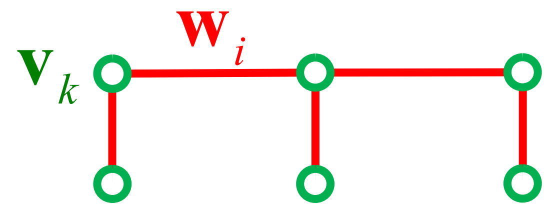
Parameters to be determined: ① state s ② action a ③ reward r







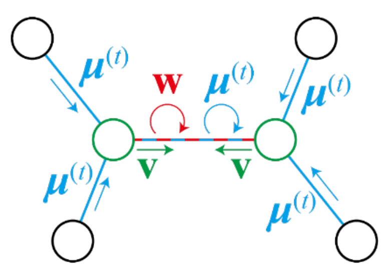
① state s



index	Member input $\mathbf{w}_i$
1	1 if column, else 0
2	1 if beam, else 0
3	(member length)/12.0
4	size index (200,250, …, 1000)
5	stress ratio
6	displacement ratio

index	Node input $\mathbf{v}_k$
1	1 if supported; 0 else
2	1 if at the top, 0 else
3	1 if at the side ends, 0 else
4	COF ratio

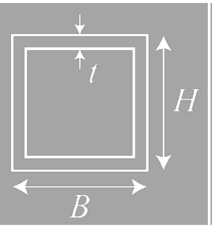
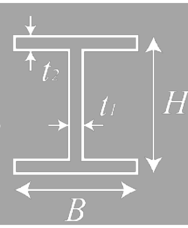
$$\{\mathbf{w}_1,\cdots,\mathbf{w}_m,\mathbf{v}_1,\cdots,\mathbf{v}_n\} \xrightarrow{\text{graph embedding}} \{\boldsymbol{\mu}_1,\cdots,\boldsymbol{\mu}_m\} \simeq s$$

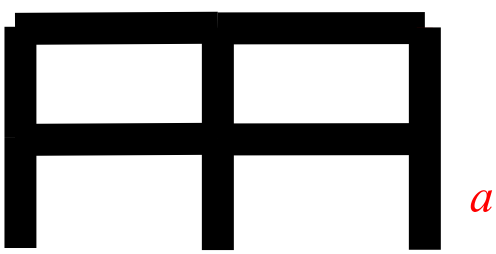


② Action a (reducing size ver.)

Action a : Reduce size index J_a by one level

(Automatically adjust above columns' size if lower column becomes more slender)

Ji	$H \times B \times t$ 	$H \times B \times t_1 \times t_2$ 
200	$200 \times 200 \times 12$	$194 \times 150 \times 6 \times 9$
250	$250 \times 250 \times 12$	$244 \times 175 \times 7 \times 11$
⋮	⋮	
900	$900 \times 900 \times 36$	$900 \times 300 \times 16 \times 28$
950	$950 \times 950 \times 36$	$950 \times 300 \times 16 \times 28$
1000	$1000 \times 1000 \times 36$	$1000 \times 300 \times 16 \times 28$



③ Reward r (reducing size ver.)

Reward $r \in$ [-1,1] : depends on the change of stress, displacement and COF

$$r = \frac{1}{3}\left(C\left(\frac{\max_i \tilde{\sigma}_i'}{\max_i \tilde{\sigma}_i}\right) + C\left(\frac{\min_i \beta_i}{\min_i \beta_i'}\right) + C\left(\frac{\max_i \tilde{d}_i'}{\max_i \tilde{d}_i}\right) \right)$$

COF

stress ratio

one-step previous values

displacement ratio

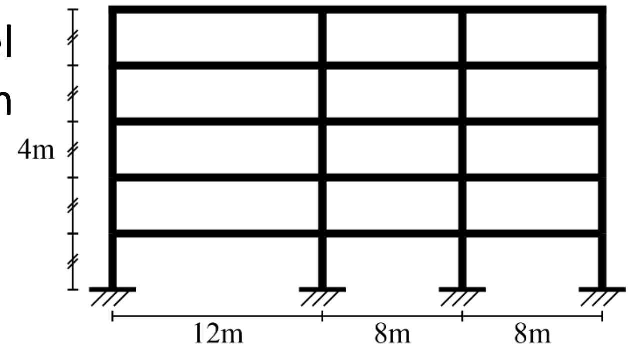
Number of size-changed members

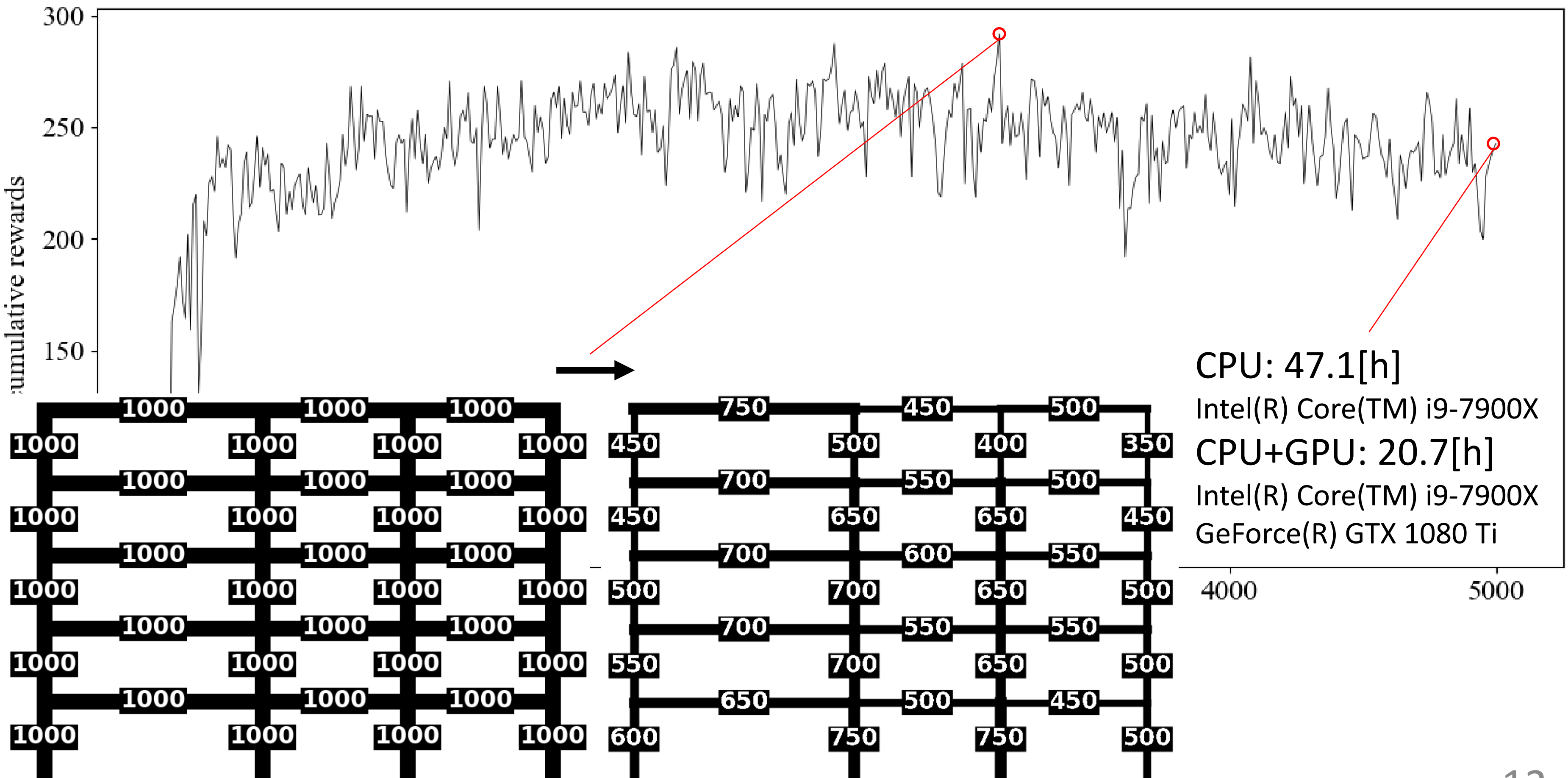
$$C(x) = \begin{cases} \min\{x, 1.0\} & \text{if (solution is feasible)} \\ 0 & \text{else if (} x \text{ satisfies constraint)} \\ \frac{n_e}{\sqrt{n_s}} \max\{-\frac{1}{x}, -1.0\} & \text{else} \end{cases}$$

Number of stories

Training result

Frame model for validation





CPU: 47.1[h]

Intel(R) Core(TM) i9-7900X

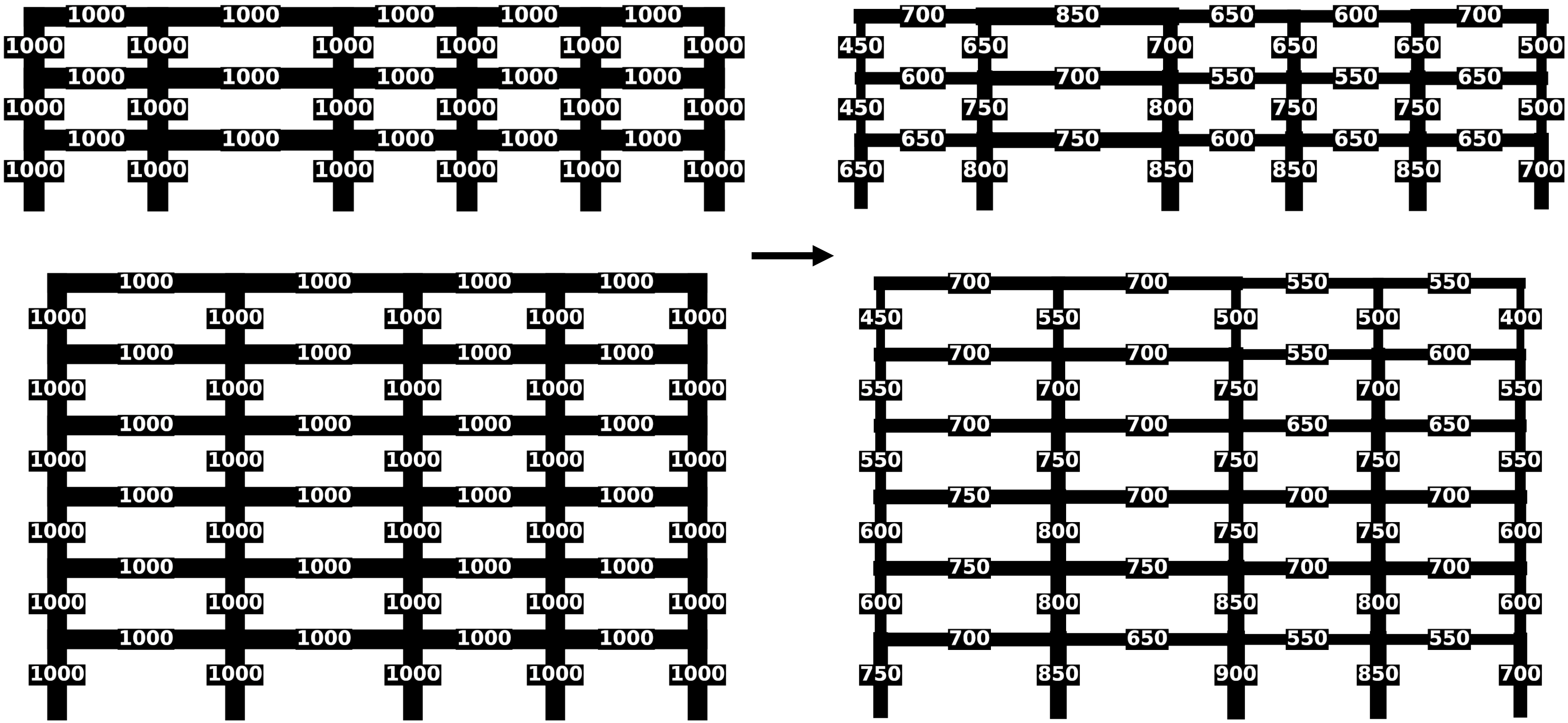
CPU+GPU: 20.7[h]

Intel(R) Core(TM) i9-7900X

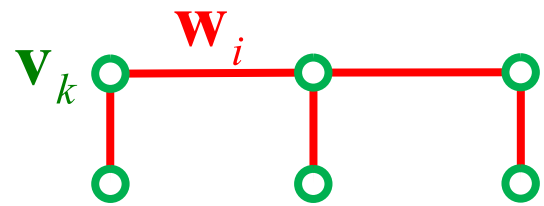
GeForce(R) GTX 1080 Ti

Applicability to different frames

Trained agents can be applied without re-training



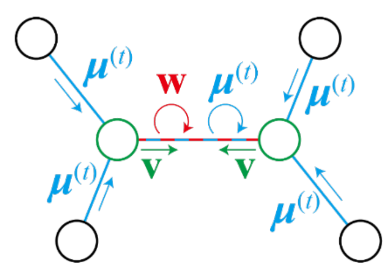
① state s



index	Member input $\mathbf{w}_i$
1	1 if column, else 0
2	1 if beam, else 0
3	(member length)/12.0
4	size index (200,250, …, 1000)
5	stress ratio
6	displacement ratio

index	Node input $\mathbf{v}_k$
1	1 if supported; 0 else
2	1 if at the top, 0 else
3	1 if at the side ends, 0 else
4	COF ratio

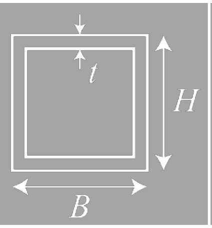
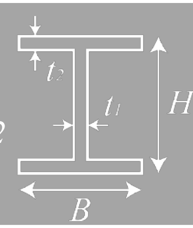
$$\{\mathbf{w}_1, \cdots, \mathbf{w}_m, \mathbf{v}_1, \cdots, \mathbf{v}_n\} \xrightarrow{\text{graph embedding}} \{\boldsymbol{\mu}_1, \cdots, \boldsymbol{\mu}_m\} \simeq s$$

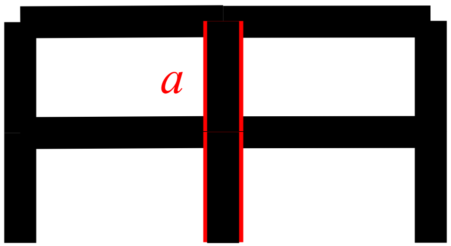


② Action a (increasing size ver.)

Action a : Increase size index J_a by one level

(Automatically adjust lower columns' size if upper column becomes thicker)

Ji	$H \times B \times t$ 	$H \times B \times t_1 \times t_2$ 
200	$200 \times 200 \times 12$	$194 \times 150 \times 6 \times 9$
250	$250 \times 250 \times 12$	$244 \times 175 \times 7 \times 11$
⋮	⋮	
900	$900 \times 900 \times 36$	$900 \times 300 \times 16 \times 28$
950	$950 \times 950 \times 36$	$950 \times 300 \times 16 \times 28$
1000	$1000 \times 1000 \times 36$	$1000 \times 300 \times 16 \times 28$



③ Reward r (increasing size ver.)

Reward $r \in$ [-1,1] : depends on the change of stress, displacement and COF

$$
r = \begin{cases} 1.0 & \text{if (feasible)} \\ \dfrac{n_e}{3\sqrt{n_s}} \left(C\left(\dfrac{\max_i \tilde{\sigma}_i'}{\max_i \tilde{\sigma}_i}\right) + C\left(\dfrac{\min_i \beta_i}{\min_i \beta_i'}\right) + C\left(\dfrac{\max_i \tilde{d}_i'}{\max_i \tilde{d}_i}\right) \right) & \text{else} \end{cases}
$$

Number of size-changed members (n_e)

Number of stories (n_s)

COF (β_i)

stress ratio ($\tilde{\sigma}_i$)

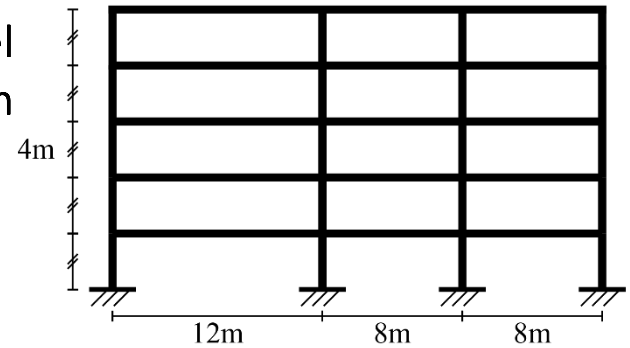
one-step previous values ($\tilde{\sigma}_i'$, β_i', $\tilde{d}_i'$)

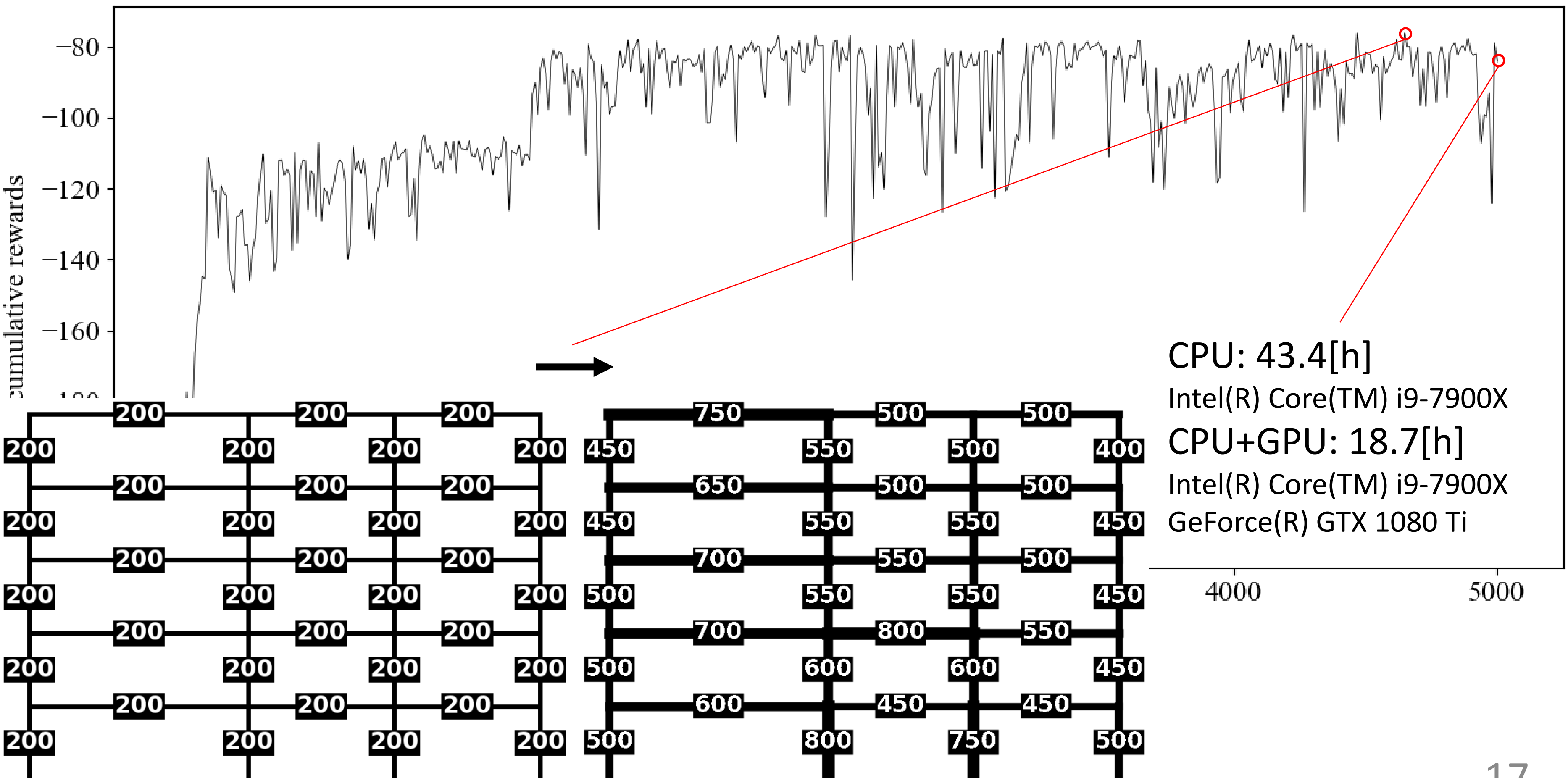
displacement ratio ($\tilde{d}_i$)

$$C(x) = \max\{x, -1.0\}$$

Training result

Frame model for validation



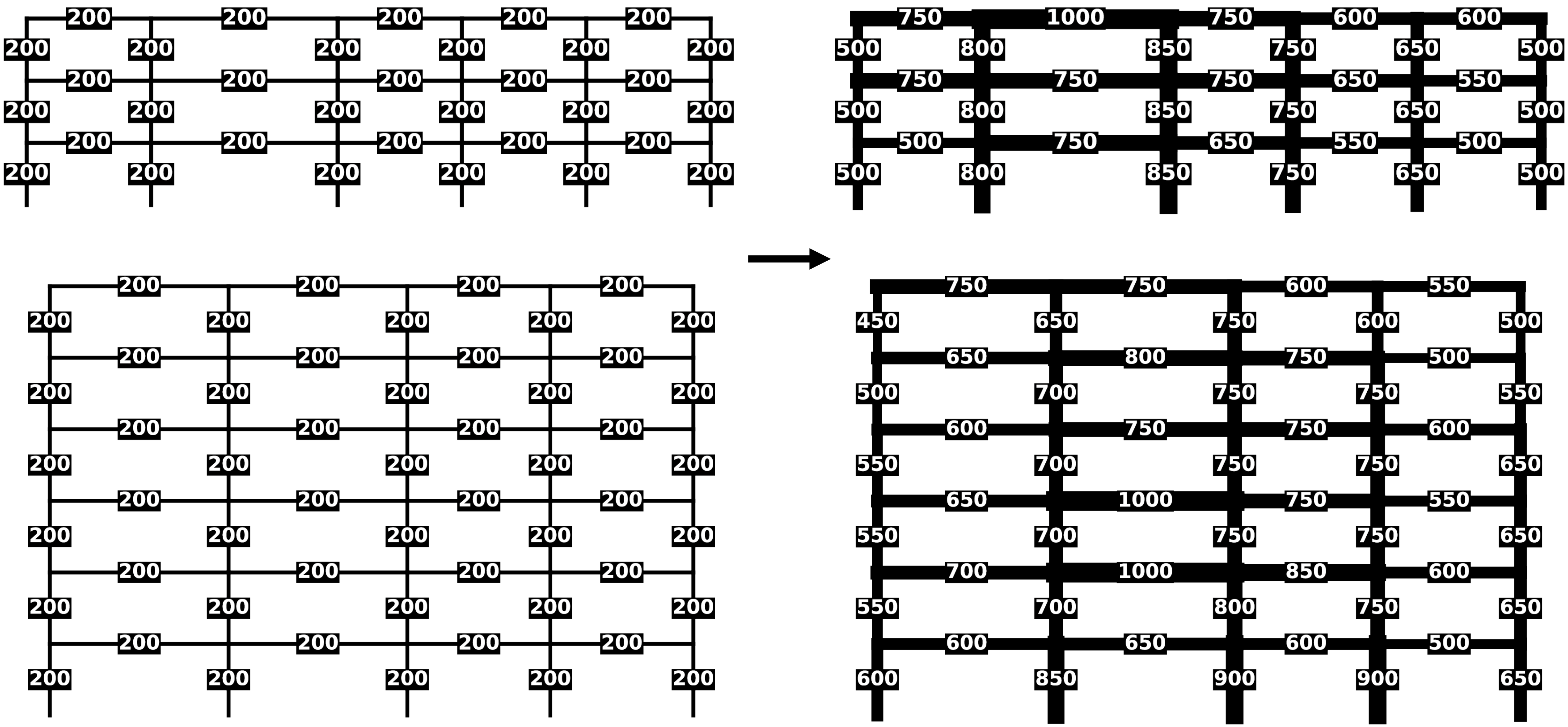


CPU: 43.4[h]
Intel(R) Core(TM) i9-7900X

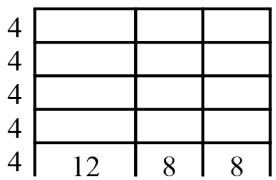
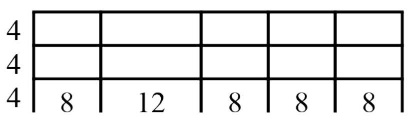
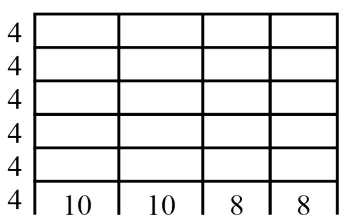
CPU+GPU: 18.7[h]
Intel(R) Core(TM) i9-7900X
GeForce(R) GTX 1080 Ti

Applicability to different frames

Trained agents can be applied without re-training



VS particle swarm optimization (PSO)

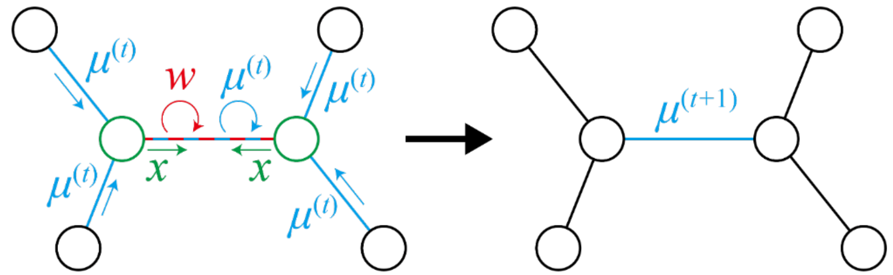
problem	RL+GE(-)	RL+GE(+)	PSO
	$t = 2.6$ $V = 6.778$	$t = 2.3$ $V = 6.418$	$t = 6.2$ $V = 6.642$
	$t = 1.7$ $V = 7.995$	$t = 2.4$ $V = 8.207$	$t = 8.0$ $V = 6.220$
	$t = 3.9$ $V = 12.940$	$t = 3.9$ $V = 13.554$	$t = 18.3$ $V = 12.999$

t : elapsed CPU time
V : structural volume

Smaller t
→ RL+GE is faster than PSO

Similar V
→ RL+GE can obtain solutions comparable to PSO

Conclusion



- A hybrid method of reinforcement learning and graph embedding is proposed for minimum-volume design of steel frames
- Trained agents are able to apply design change to members considering constraints and objectives of the structural design problem
- The trained agent can be applied to various structures at a low computational cost regardless of the number of nodes and members and shape

contact: hayashi.kazuki.55a@st.kyoto-u.ac.jp