Synergetic-informed deep reinforcement learning for sustainable management of transportation networks with large action spaces

Lai, Li; Dong, You; Andriotis, Charalampos P.; Wang, Aijun; Lei, Xiaoming

DOI
10.1016/j.autcon.2024.105302

Publication date
2024

Document Version
Final published version

Published in
Automation in Construction

Automation in Construction 160 (2024) 105302

Contents lists available at ScienceDirect

Automation in Construction

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

Synergetic-informed deep reinforcement learning for sustainable management of transportation networks with large action spaces

Li Lai [a], You Dong [a], Charalampos P. Andriotis [b], Aijun Wang [c], Xiaoming Lei [a,*]

[a] Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hong Kong, China
[b] Faculty of Architecture and the Built Environment, Delft University of Technology, the Netherlands
[c] Wuhan University of Technology, China

A B S T R A C T

Effective transportation network management systems should consider safety and sustainability objectives. Existing research on large-scale transportation network management often employs the assumption that bridges can be considered individually under these objectives. However, this simplification misses accurate system-level representations, induced by multiple components, network topology, and global maintenance actions. To address these limitations, this paper presents a deep reinforcement learning (DRL) framework that draws inspiration from biological learning behaviors to determine optimal life-cycle management policies. It incorporates synergetic branches and hierarchical rewards, factorizing the action space and, thereby, diminishing system complexity from exponential to linear with respect to the number of bridges. Extensive experiments based on a realistic case study demonstrate that the proposed method outperforms expert maintenance strategies and state-of-the-art decision-making methods. Overall, the proposed DRL framework can assist engineers by offering adaptive solutions to maintenance planning. It also provides solutions that address large action spaces within complex systems.

Keywords
Deep reinforcement learning
Infrastructure management
Maintenance optimization
Hierarchical reward
Life-cycle analysis
Large discrete action spaces

1. Introduction

Transportation networks are of utmost importance for economic development and population mobility. Nevertheless, the performance deterioration of transportation networks is inevitable due to external loads, material aging, and environmental corrosion, especially for bridges which are the most vulnerable and costly components of the system. In the USA, approximately a quarter of bridges are either structurally deficient or functionally obsolete [1]. The investment shortfall for the transportation network will be $549.5 billion in the next five years. Moreover, with the additional requirements in serviceability, transportation systems need to satisfy sustainable requirements, such as economic and societal [2,3]. Carbon neutrality is also becoming more important, and China aims at realizing the CO_2 emission inflection point before 2035 [4,5]. Based on the results by [4], the total CO_2 emission in 2020 was approximately 34 billion tons, with vehicles occupying around 10% ~ 15% of emissions. A considerable portion of emissions is generated by vehicle detours caused by bridge maintenance, which can be reduced through transportation network management strategies. These additional requirements make the existing transportation network management system, which only focuses on bridge safety [6–8], no longer suitable. Therefore, a future study in decision-making for transportation network management has to explicitly include consideration of various constraints in the optimization to satisfy the sustainability goals, including reducing greenhouse gas emissions, promoting energy efficiency, and enhancing public transportation. In the transformative age of automation and artificial intelligence (AI), a dedicated AI-aided decision-making system should be developed to utmost extent address the complex transportation network management problem. This system would be invaluable in addressing the challenges of managing transportation networks, especially considering the multifaceted sustainability requirements, diverse information fusing, and the need for optimal operation under resource constraints.

Effective management systems are essential for decision support and are used throughout the whole life-cycle of the transportation networks, from cradle to grave [9–11]. For this, many management algorithms have been developed to consider the reasonable timing, type, and extent of the interventions based on the inspection data, structural condition,

* Corresponding author.
E-mail address: xiaoming.lei@polyu.edu.hk (X. Lei).

https://doi.org/10.1016/j.autcon.2024.105302
Received 17 April 2023; Received in revised form 28 January 2024; Accepted 28 January 2024
Available online 6 February 2024

maintenance facility deployment, and geographical distribution [12–14]. In the early stage, the studies mainly focus on the optimal maintenance planning for individual bridges in transportation networks. Various reliability-based frameworks are developed to describe the bridge time-dependent deterioration process and quantify the uncertainty in transportation network management [15]. Then, due to the highly self-adaptive capability, genetic algorithms (GAs) were employed to optimize on an annual basis the maintenance efforts for a concrete bridge over the life cycle [16,17]. However, the fundamental procedure of GAs involves encoding the state of the infrastructure as an 'individual' and constructing a 'population' [18], where each 'individual' is quantified by a set of Genes represented in binary (i.e., 0 or 1). It indicates with the number of bridges in the system increasing, the number of genes that need to be defined will grow exponentially. To avoid the complicated encoding and large-scale state representations [19,20], for complex system management with GAs, certain simplifications and assumptions are needed to design for transportation systems [3,21].

Another limitation of GAs is the optimized efficiency. GAs often find acceptable rather than optimal or near-optimal solutions. This is demonstrated in chess games, where players trained through Genetic Algorithms (GA) may attain a rating of 1600 [22], whereas, in comparable time frames, those trained using Deep Reinforcement Learning (DRL) can achieve a significantly higher rating of 3500 [23]. In addition, DRL has been shown to outperform GAs in sequential decision-making problems [22] and in managing high uncertainty problems [23]. Based on the characteristics of transportation systems, such as large state and action spaces and high uncertainty in system evolution, we decide to use DRL to optimize maintenance policies. We primarily adopt the Markov model in DRL, as well as the bottlenecks that prevent DRL from being applied in complex infrastructure management.

Markov Decision Processes (MDPs) are commonly employed in infrastructure management to quantitatively model the life-cycle behavior of a system, which consists of a set of states, actions, and rewards. It offers a robust mathematical framework for closed-loop control, particularly for optimal sequential decision-making in discrete-time and discrete-state scenarios. Partially observable Markov decision processes (POMDPs) are an extension of MDPs that address the limitations of complete information by allowing for partial observability [24,25]. A key aspect contributing to the uncertainty consideration of POMDPs is their utilization of a belief state, representing the probability distribution over the possible states of the structure. As new observations (e.g., inspection and monitoring) are made, the belief state is updated using Bayes' theorem, allowing the model to adapt to new information and make decisions based on the most up-to-date knowledge of the system's state [26]. In addition, this framework allows the policy-making problem to scale flexibly with the number of decision steps, states, and actions [26]. The belief state provides a compact representation of the infrastructural uncertainty, allowing for efficient computation and decision-making by considering the probability distribution over states rather than exhaustively analyzing each state individually. This flexibility enables POMDPs to quantify complex system management that involves multiple components as well as long planning horizons [27].

In this regard, there is growing interest in implementing POMDPs in condition control of infrastructure assets, for instance, the highway pavement [28], bridge components [29], and blades in a wind farm [30]. However, it is worth noting that the studies mentioned above are restricted to the component-level because the traditional solving algorithms for POMDPs are point-based iterations [31]. These algorithms lose their edge when the system state and action combinations scale exponentially with the number of components considered in large-scale systems [32]. Therefore, other efficient methods are necessary to deal with the computational complexity that severely grows in large-scale infrastructure management.

The large-scale state and action number in complex systems management has motivated the application of Deep Reinforcement Learning (DRL) techniques [33,34]. DRL can handle large-scale state value points by learning representations of the belief state (input) and state-value (output) through neural networks [35]. Additionally, DRL's online learning capabilities allow it to adapt to changing conditions, continually improving its performance over time and addressing the challenges associated with large state and action spaces in evolving environments [35]. Moreover, DRL offers unprecedented capabilities in providing near-optimal solutions to a series of complex planning and decision-making tasks, outperforming humans in fields traditionally dominated by experts [36–38]. It has opened a novel path in how to model and control complicated systems with high-dimensional inputs and outputs, extract the characteristics from multifarious structural responses, and demystify the complex governing system or virtual entities (e.g., digital twins). DRL is essentially the process of learning the optimal policy by interacting between the agent and the MDPs-based and POMDPs-based environment.

Deep *Q*-Network (DQN) in DRL, as a surrogate model of the value function, has been widely to keep the computational effort acceptable. DQN takes state information as input and outputs corresponding actions and state-action values, as shown in Fig. 1. The large-scale state-action pairs are approximated using neural networks which save a significant amount of memory space. This improvement motivates infrastructure management to develop from the component-level to the structural-level, such as from beams and piers to cable-stayed bridges [39], from pavement to highway [40], and other multi-component systems [41].

Although DQN can consider the states of inhomogeneous components in a complex system, the neural network is required to output the *Q*-value of every possible maintenance action combination. Without any assumptions to simplify the synergy of different components' action combinations, the number of nodes in the output layer would increase exponentially with the number of components considered. To avoid the sizable neural network, the synergy of maintenance actions often has to be neglected for DQN. However, this simplification is contrary to the requirements of transportation network management since a small maintenance change can induce cascading effects on the entire system. For instance, the interruption of an arterial road due to bridge rehabilitation will dramatically affect the traffic flows which increase the detour and reduce the mobility of adjacent lanes.

Dealing with large-scale discrete action spaces is an important active area of research in DRL. Dulac-Arnold et al. [42] developed the Deep Deterministic Policy Gradient (DDPG) algorithm with the Wolpertinger architecture, which enables DDPG to be applied in large-scale discrete action control problems. The Wolpertinger architecture utilizes prior information about the discrete actions to embed them into the continuous space. Tavakoli et al. [43] modified dueling DQN [44] with action branching architectures (BDQN) to consider the synergistic effect of different action dimensions. Chen et al. [45] provided a tree-structured policy gradient recommendation (TPGR) framework which uses the leaf to consider entire discrete actions and non-leaf nodes as the path-chosen unit. We reproduced all the methods above and applied them in transportation network management, but the outcomes were unsatisfactory. Approaches like DDPG with Wolpertinger (DDPG-W), break the consistency and differentiability of the neural network, potentially resulting in improper backpropagation of gradients and instabilities in training. For example, the continuous action of the control steering wheel and joint can be discretized into serval fixed rotation angles. Through interacting with the environment, positive or negative feedback can guide correct gradients to adjust the rotation angles. However, for bridge management in the transportation network, this feedback lacks intuitive physical significance for different maintenance scenarios and cannot direct the neural network to choose optimal actions. Other methods, such as BDQN and TPGR, using a single loss function cannot effectively deliver the complex information from the environment and readily converge to sub-optimal solutions. Therefore, existing research is not sufficiently competitive in managing complex transportation network, especially when considering the multi-feature of the system (e.g., bridge deterioration, road grade, and traffic flows), multiple

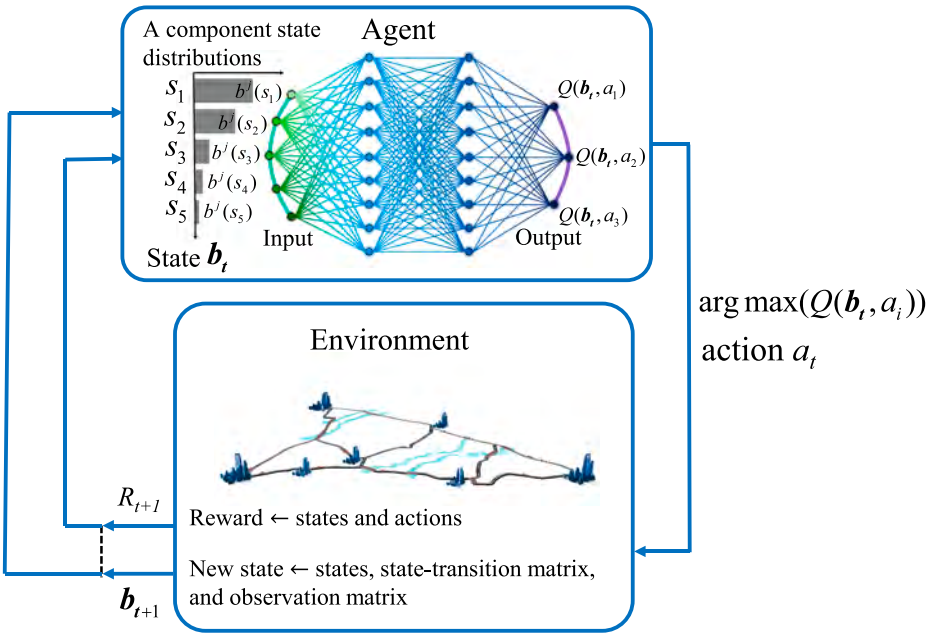

Fig. 1. Schematic diagram of DRL.

objectives optimization (e.g., safety, carbon neutrality, and mobility), and synergy effects between different bridge maintenance actions. But past and recent advances in the AI field have opened new paths in how we can learn to model and control complex systems from noisy and high-dimensional real-world or synthetic inputs and outputs and how we can quantify information from multiple data resources. This is to say that, in this stage, a dedicated AI-aided decision-making algorithm is possible and necessary to design which will perform competitively in complex transportation network management.

Along these lines, firstly, to maximize the quantification of a large transportation network composed of multiple infrastructure components, the POMDPs model simulates all infrastructure in the system without losing any features. These features include uncertainties of bridge deterioration processes, inspection errors, uncertainties in the improvement of structural performance through maintenance actions, traffic flow redistribution caused by bridge state and maintenance actions, and additional carbon emissions induced by detours. Unlike other models, this model does not consider bridges individually, but rather takes into account the interdependence between each maintenance action. In addition, to consider multiple attributes of the transportation system such as safety, sustainability, mobility, and life-cycle cost, a multi-objective function is designed for this complex system management. If each bridge's available maintenance actions are regarded as a separate dimension, then the maintenance actions for the entire transportation system would form discrete points in a high-dimensional space. To obtain the optimal maintenance policy, we leverage the expressive capability of BDQN in high-dimensional action space and develop a cutting-edge training algorithm, favorably tailored to finely control and management of large engineering systems associated with high uncertainty. The proposed training method is inspired by the learning process of creatures, starting from simple non-conditioned reflexes to complex conditioned reflexes. More specifically, the hierarchical multi-reward backpropagation learning mechanism is developed to thoroughly consider the individual-level and system-level feedback for the refined control in the complicated system. Through the comparative studies in real-world transportation system management, the proposed framework has outperformed other DRL methods, and the maintenance policy recommended by the framework has proven to be superior to those suggested by experienced engineers in infrastructure management.

In summary, the following innovations and contributions are proposed: 1) the complicated real-world transportation network management problem is quantified as POMDPs which can consider high uncertainty in transportation network evolution and multiple objectives among stakeholders; 2) a versatile decision-making framework is constructed to deal with the large discrete action space in the complex engineering system management; and 3) a novel neural network training method is proposed in which the training mechanism imitates the animal learning behavior from simple tasks to complex assignments. Overall, these contributions advance the state-of-the-art in transportation network management and have significant implications for ensuring transportation networks' safety, functionality, and sustainability.

2. POMDPs-based deep reinforcement learning

2.1. Partially observable Markov decision processes

MDPs provide a mathematical framework to quantify sequential decision-making problems in stochastic environments. The agent in MDPs will interact with the defined environment in finite time steps with observations and actions. POMDPs additionally consider the uncertainty in system condition assessment and error in observation. To visually illustrate these concepts, a definition graph of POMDPs associated with transportation network is shown in Fig. 2. POMDPs can be quantified by 7-tuple parameters $E = \langle \mathbb{S}, \mathbb{A}, \mathbb{O}, \boldsymbol{T}, \boldsymbol{\Theta}, \boldsymbol{R}, \gamma \rangle$. The first three hollow capital letters indicate the sets of bridge states ($\mathbb{S}$), maintenance actions ($\mathbb{A}$), and bridge inspection value ($\mathbb{O}$), respectively. The corresponding lowercase symbols represent the discrete elements in those sets, e.g., bridge states are defined as $s_i \in \mathbb{S}$ based on the standard JTG/T H21–2011 [46]. The number of elements in the set is denoted by scalar symbols, $|\mathbb{S}|$, $|\mathbb{A}|$, and $|\mathbb{O}|$. In this paper, bold letters are used to represent matrices or vectors. Since the uncertainty of structural condition assessment is unavoidable, j^{th} bridge condition is expressed as the probabilistic distribution over the possible states, $\boldsymbol{b}^j = (b(s_1), b(s_2), \ldots, b(s_n))$, and entire system condition is represented by a vector, $\boldsymbol{b} = \left(\boldsymbol{b}^1, \boldsymbol{b}^2, \ldots, \boldsymbol{b}^m\right)$, where m is the number of bridges in the system and n is the state number. A three-dimensional state-transition matrix $\boldsymbol{T}$ ($|\mathbb{S}| \times |\mathbb{S}| \times |\mathbb{A}|$) characterizes the state transition probability

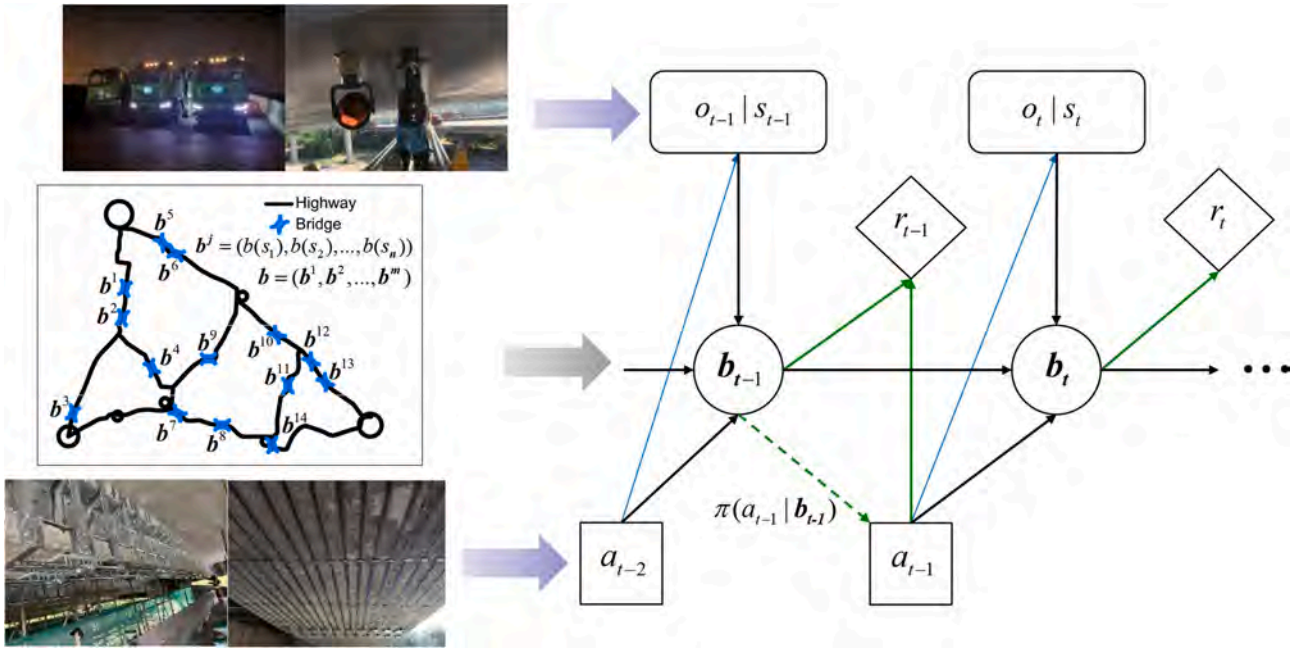

Fig. 2. Probabilistic graphical model of POMDPs.

($p(s_t|s_{t-1}, a_{t-1})$) when maintenance action a_{t-1} is conducted at state s_{t-1}. The state-dependent observation probability matrix $\boldsymbol{\Theta}$ ($|\mathbb{O}| \times |\mathbb{S}| \times |\mathbb{A}|$) defines the probability of inspected value $p(o_t|s_t, a_{t-1})$ after the maintenance action a_{t-1} is executed and a bridge state is transferred to s_t. Reward matrix $\boldsymbol{R}$ ($|\mathbb{S}| \times |\mathbb{A}|$) defines the immediate cost $r(a_t, s_t)$ when the agent conducts an action a_t at state s_t. Parameter γ quantifies the discount factor over the time horizon.

In Fig. 2, in an arbitrary time step $t-1$, the state of the transportation network is expressed as $\boldsymbol{b}_{t-1}$. After the agent executing a maintenance action a_{t-1} and receives an inspected value o_t, the agent will obtain the posterior state distribution $p\left(\boldsymbol{b}_t^j|o_t, a_{t-1}, \boldsymbol{b}_{t-1}^j\right)$ of the j^{th} bridge on the basis of a Bayesian updating [31]:

$$b_t^j(s_t) = p\left(s_t|o_t, a_{t-1}, \boldsymbol{b}_{t-1}^j\right) = \frac{p(o_t|s_t, a_{t-1})}{p\left(o_t|a_{t-1}, \boldsymbol{b}_{t-1}^j\right)} \sum_{s_t \in \mathbb{S}} p(s_t|s_{t-1}, a_{t-1}) b^j(s_{t-1}) \tag{1}$$

where the denominator, $p\left(o_t|a_{t-1}, \boldsymbol{b}_{t-1}^j\right)$, is the standard normalization coefficient, calculated by:

$$p\left(o_t|a_{t-1}, \boldsymbol{b}_{t-1}^j\right) = \sum_{s_t \in \mathbb{S}} p(o_t|a_{t-1}, s_t) \sum_{s_{t-1} \in \mathbb{S}} p(s_t|s_{t-1}, a_{t-1}) b^j(s_{t-1}) \tag{2}$$

The black arrows in Fig. 2 correspond to Eq. (1) and (2), which unveils the bridge state transition path in POMDPs, given as:

$$p\left(\boldsymbol{b}_t^j|a_{t-1}, \boldsymbol{b}_{t-1}^j\right) = \sum_{o_t \in \mathbb{O}} p\left(o_t|a_{t-1}, \boldsymbol{b}_{t-1}^j\right) \tag{3}$$

It is worth noting that there are two assumptions in POMDPs. The continuous deterioration processes are approximately represented by the finite discrete state changing. Another assumes that the current belief state ($\boldsymbol{b}_t$) and selected action (a) include sufficient information to determine the next belief state ($\boldsymbol{b}_{t+1}$), regardless of the historical states and action sequences. Once those two assumptions are acceptable, the performance of the transportation network is easily transformed into a Markovian process.

The green arrows in Fig. 2 define how the agent chooses sequential actions to maximize the reward or minimize the maintenance cost in the life-cycle. The state-dependent parameter, policy $\pi(a_t|\boldsymbol{b}_t)$, is introduced to guide the agent's decision-making based on the current belief state. In each time step, the state will map to a specific action based on the policy: $\pi(\boldsymbol{b}_t) \rightarrow a_t$, as shown the dashed green line in Fig. 2. Then, in the life-cycle of the transportation network, the total maintenance fee guided by this policy is calculated by:

$$\begin{aligned} V^{\pi} &= \underset{\pi \rightarrow a}{\boldsymbol{E}}\left[\sum_{t=1}^{L} \gamma^{t-1}\left(\sum_{j=1}^{m}\sum_{i=1}^{n} r_j(s_i, a_t) b_t^j(s_i) + r_E(a_t, \boldsymbol{b}_t)\right)\right] \\ &= \underset{\pi \rightarrow a}{\boldsymbol{E}}\left[\sum_{t=1}^{L} \gamma^{t-1} r(a_t, \boldsymbol{b}_t)\right] \end{aligned} \tag{4}$$

where L refers to the lifespan of the system. Similar to Eq. (1), $b_t^j(s_i)$ in Eq. (4) represents the probability of j^{th} bridge condition belonging to state s_i at time step t. The total reward consists of the maintenance fee of individual bridge $\sum_{i=1}^{n} r_j(s_i, a_t) b_t^j(s_i)$ and feedback (e.g., CO_2 emission and mobility) from the entire transportation network $r_E(a_t, \boldsymbol{b}_t)$. To simplify the eq. (4), the entire value is abbreviated as $r(a_t, \boldsymbol{b}_t)$. Value Eq. (4) plays a vital part in the decision-making of transportation network management. The objective of management is to plan sequential actions to maximize this value V^{π}.

To obtain the optimal policy $\pi^*(\boldsymbol{b}) \rightarrow a$ which can guide the agent to choose the best action in any belief state, Eq. (4) is rewritten compactly to a one-step forward pattern based on the current cost and expected future reward. This function is named the Bellman Eq. [47]:

$$V^{\pi^*}(\boldsymbol{b}_t) = \max_{a_t \in \mathbb{A}}\left[r(a_t, \boldsymbol{b}_t) + \gamma \sum_{o_{t+1} \in \mathbb{O}} p(o_{t+1}|a_t, \boldsymbol{b}_t) V^{\pi^*}(\boldsymbol{b}_{t+1}|o_{t+1}, a_t, \boldsymbol{b}_t)\right] \tag{5}$$

In Eq. (5), the first item refers to the instantaneous reward which can directly get from parameter $\boldsymbol{R}$. The second item is the expected belief point value $V^{\pi^*}(\boldsymbol{b}_{t+1})$ after the bridge state is transferred. An important mathematical feature of the $V^{\pi^*}(\boldsymbol{b})$ in POMDPs is convex and piecewise linear, which $V^{\pi^*}(\boldsymbol{b})$ can be approximated by multiple linear polynomial functions [47] or neural networks [48]. For the previous one, the belief state value is calculated by the point-based algorithm, given as:

$$V^{\pi^*}(\boldsymbol{b}) = \max_{\alpha \in \Gamma} \sum_{j=1}^{m}\sum_{i=1}^{n} \alpha^j(s_i) b^j(s_i) \tag{6}$$

To better understand Eq. (6), each parameter is explained with geometrical meaning. If a belief state ($\boldsymbol{b}$) is regarded as a point in $n \times m$

dimensional space, then the value coefficient ($\boldsymbol{\alpha}$) is the gradient of this point. Because of the property of piecewise linear, the adjacent belief points will share the same gradient ($\boldsymbol{\alpha}$). As a result, the entire belief state value can be precisely represented by finite hyperplanes which are calculated by the cluster of value coefficients $\boldsymbol{\Gamma} = \{\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_i\}$. Since each vector ($\boldsymbol{\alpha}_i$) is also associated with a specific maintenance action [49], thus, the computation of the optimal policy of POMDPs is to obtain a suitable cluster Γ which can be determined by point-based algorithms. A detailed description of the point-based method can be found in Shani et al. [50]. Herein, a brief introduction is given, and we will highlight the reason why this approach cannot tackle large-scale system management. In the point-based algorithms, a key step is to sample a set of representative belief points based on the value boundary method [51]. Then, the value coefficients ($\boldsymbol{\Gamma}$) for belief points are calculated by iteration of the Bellman function (5). However, due to the large number of state and action combinations in the transportation network, the number of belief points and computation of value-iteration exponentially increase with the bridge number. Thereby, point-based algorithms can only be implemented in small-to-medium-sized infrastructure management.

To this end, DRL as a surrogate value model can alleviate the complexity of computation. It uses the transportation network states ($\boldsymbol{b}$) as the input vector in neural networks and outputs the belief state value ($V(\boldsymbol{b})$). The point-based belief state is expressed as the input vector in neural networks which reduces the complexity from $|\mathbb{S}|^m$ to $|\mathbb{S}| \times m$. In addition, the value coefficient vectors ($\boldsymbol{\Gamma}$) for the belief state in the point-based algorithm are also simplified by output in neural networks. Therefore, transportation network management in conjunction with DRL is indispensable to accommodate the large state and action space.

2.2. Deep reinforcement learning

In neural network training, the difference between the objective and the output, which is called the error or loss value, is used as the back-propagation. The kernel of DRL is an 'evolving' model in that this difference dynamically changes with interaction with the environment. Compared to other neural network training methods or network types, the construction of temporal difference learning is a central part of DRL. Those approaches can be generally categorized into two major groups: on-policy learning and off-policy learning [52]. The on-policy indicates that the agent will successively improve the current policy when interacting with the defined POMDPs in a given environment, e.g., the Actor-Critic algorithm in [53]. Such networks will generate the probabilistic distribution of executing each maintenance action. Compared to DQN, where maintenance actions are determined deterministically, theoretically, Actor-Critic methods are more suitable for constrained POMDPs and other more complex environments where the optimal action can often be a probability distribution. Training networks in the form of Actor-Critic for problems with large action spaces can invoke similar issues with DQN in relation to large action spaces, unless multi-agent assumptions are employed [54]. The off-policy refers to the agent updating the current policy while executing a different policy in the environment, e.g., DQN and DDPG. Off-policy algorithms store their experiences $\langle \boldsymbol{b}_t, a_t, r_t, \boldsymbol{b}_{t+1} \rangle$ in the replay buffer $U(D)$ and keep policy consistency ($\pi(\boldsymbol{b}) \rightarrow a$) in the next few time steps. Then, the neural network is updated every fixed number of time steps. Due to the randomness in state transition and error in observation, the replay buffer in off-policy learning can serve training stability and robustness [36]. Therefore, this study concentrates on off-policy learning algorithms.

2.3. Deep Q-networks

Since the proposed framework is developed from DQN, the pertinent algorithmic and mathematical formulations will be primarily discussed. To accommodate the structure of the neural networks, the Bellman optimality Eq. (5) in POMDP is adjusted. The parameter state-action value ($Q(\boldsymbol{b}, a)$) is introduced [55], given as:

$$Q(\boldsymbol{b}_t, a_t) = r(\boldsymbol{b}_t, a_t) + \gamma \bullet \underset{a \in \mathbf{A}}{max} Q(\boldsymbol{b}_{t+1}, a_{t+1}) \tag{7}$$

Where Q is defined as the value when the agent adopts action a_t at belief state $\boldsymbol{b}_t$. Herein, a single bridge maintenance problem is adopted as a paradigm to introduce DQN. The structure of DQN is shown in Fig. 3, in which the belief state ($\boldsymbol{b}_t^j$) is introduced as input, with an appropriate number of hidden layers, and finally output the state-action value $Q\left(\boldsymbol{b}_t^j, a_{t+1}; \theta\right)$ where θ is the parameters of neural networks. The optimal action for the current belief state is determined by selecting the node in the output layer which gives the maximum Q-value. To train the neural networks (θ), the difference between feedback from the environment and DQN estimation is utilized as loss value in the back-propagation process, given as [55]:

$$L(\theta) = \boldsymbol{E}_{\left\langle \boldsymbol{b}_t^j, a_t, r_t, \boldsymbol{b}_{t+1}^j \right\rangle \sim U(D)} \left[r_t + \gamma \bullet \underset{a_{t+1}}{max} Q\left(\boldsymbol{b}_{t+1}^j, a_{t+1}; \theta\right) - Q\left(\boldsymbol{b}_t^j, a_t; \theta\right) \right]^2 \tag{8}$$

However, due to the maximization value step in eq. (7), the over-estimated state-action value is a popular problem in Q-learning algorithms [56]. This unrealistically over-estimated value will mislead the DQN to explore in non-optimal maintenance action space and asymptotically fall into sub-optimal policy [57]. To obtain the unbiased estimation of state-action value and stable training processes, double DQN decomposes the maximization operation in Eq. (8) into maintenance action selection and state-action value evaluation [56]. Keeping maintenance policy constant in defined training steps to construct a relatively stable environment, and update state-action value evaluation to obtain the low biased estimation. This method not only yields precise Q value estimation but brings better results in practical problems. The specific approach of double DQN is using two asynchronous updating networks in training. One is named target networks (θ^-), and another as main networks (θ). As mentioned, the double DQN decouples the action selection and Q-value estimation, the action is decided by the main networks, but value estimates are calculated by immediate reward and target networks' state-action value. As a result, Eq. (8) is adjusted to:

$$L(\theta) = \boldsymbol{E}_{\left\langle \boldsymbol{b}_t^j, a_t, r_t, \boldsymbol{b}_{t+1}^j \right\rangle \sim U(D)} \left[r_t + \gamma \bullet \underset{a_{t+1}}{max} Q\left(\boldsymbol{b}_{t+1}^j, a_{t+1}; \theta^-\right) - Q\left(\boldsymbol{b}_t^j, a_t; \theta\right) \right]^2 \tag{9}$$

In the learning process, the parameters of the main networks (θ) are updated in every time step, whereas the target networks (θ^-) follow the main networks in a slower fashion, substituting parameters $\theta^- = \theta$ with an appropriate delay.

For small- to medium-scale systems, double DQN is competitive in challenging decision-making problems with immense input spaces that were unthinkable a few years back. Nevertheless, for policy-making in systems with multiple dissimilar and interdependent constituents, the scale of the combination of actions will exponential growth with the number of components $|\mathbb{A}| = |a|^m$. Since the output layer needs to produce a Q-value for every available action combination, a sizable neural network is unavoidable in the refinement management of transportation networks. Apparently, the structure of DQN is intractable to tackle the large discrete action spaces problem. To alleviate the curse of dimensionality in the action domain, the bionic neural networks for large-scale action spaces are proposed without using simplified, less accurate modeling approaches to reduce complexity.

3. BDQN with hierarchical multi-reward backpropagation method

This section will comprehensively introduce the concept of branching dueling Q-networks (BDQN) proposed by Tavakoli et al. [43]. However, several replicated results have demonstrated that this method is prone to get trapped in local optima, even when given massive training. Then, an explanation from a theoretical perspective will be

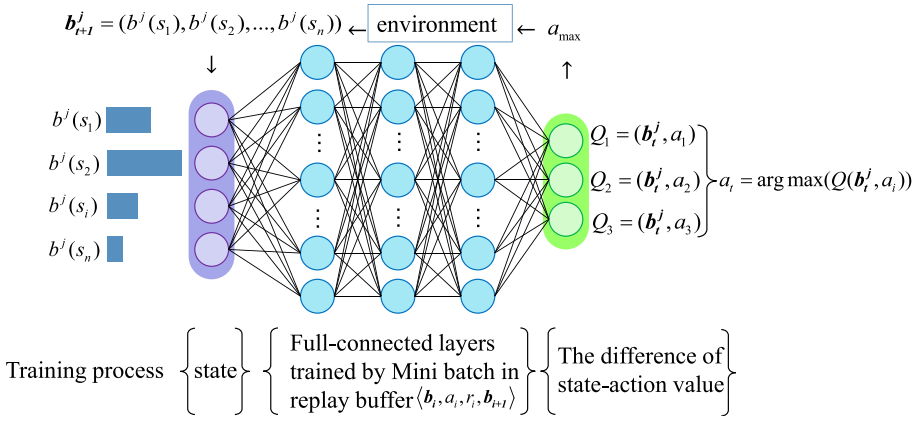

Fig. 3. Schematic diagram of DQN.

given as to why relying solely on BDQN cannot achieve satisfactory results. To this end, a hierarchical multi-reward backpropagation method comprehensively updating different parts parameters of the neural networks is proposed, which is the main contribution of the algorithm in this study.

3.1. Synergetic-informed branching dueling Q-networks

The concept of BDQN comes from bionics which simulate the behavior of Cephalopoda, such as octopus [58]. A significant number of neurons in this creature are distributed throughout its body, particularly in the arms [59]. This feature is due to the multiple arms with unlimited degrees of freedom that need to be controlled simultaneously. To alleviate the burden of the neural center, the partial delegation of control to the arms is necessary, which can decrease the response time to external impacts. The neural center is mainly responsible for coordinating the independent arm motion across its several networks. Therefore, the architecture of BDQN is designed as in Fig. 4. This neural network is divided into two parts based on functions, shared architecture simulating the 'brain' and branching architecture representing the 'arm'. When belief states are transferred into the input, the shared part layers (brain) compute a latent representation used to evaluate the state value. The branching neural network (arm) is not a fully connected layer, it factorizes belief-state-action advantages on the subsequent independent branches. Each branch manages a specific bridge in the transportation network. The number of output nodes in each branching equals available maintenance actions for the corresponding bridge. Finally, an integrated layer aggregates the state value and the factorized advantages, to output Q-value for each branching.

The scale of each branching neural network is enough to address the nonlinear for individual bridge life-cycle management. If not considering the interactive effect between maintenance actions in a transportation network, this management problem can be represented by multiple DQNs. However, the naive distribution of large multi-component systems across different independent function approximators will lead to distortion results [60]. Hence, the shared decision module with more complex hidden layers is designed to coordinate the semi-independent branches and consider the optimal solution at the system level. The shared network layers function as a 'brain' and encode the bridge's belief state information into a shared representation layer.

In order to better control the parallel branching network, the structure of dueling DQN [44] is resorted to affect the Q-value as Fig. 4 shows. The important concept in dueling DQN, named advantage function, is introduced to evaluate how advantageous an action is to the system in its current state, as defined [44]:

$$A(\boldsymbol{b}_t, a_t) = Q(\boldsymbol{b}_t, a_t) - V(\boldsymbol{b}_t) \tag{10}$$

The dueling architecture can sensitively identify the action utility and generalize more efficiently because the advantage function can distinguish between positive and negative values in mathematics. The positive value directly indicates that this action benefits the system, which guides the gradient direction in updating. To establish the dueling structure in BDQN, a particular branch is designed to estimate the current state value $V(\boldsymbol{b}_t)$. Each advantage item $A(\boldsymbol{b}_t, a_i)$ is subtracted from the mean value to highlight the positive and negative. An aggregating layer is embedded before the output layer as follows:

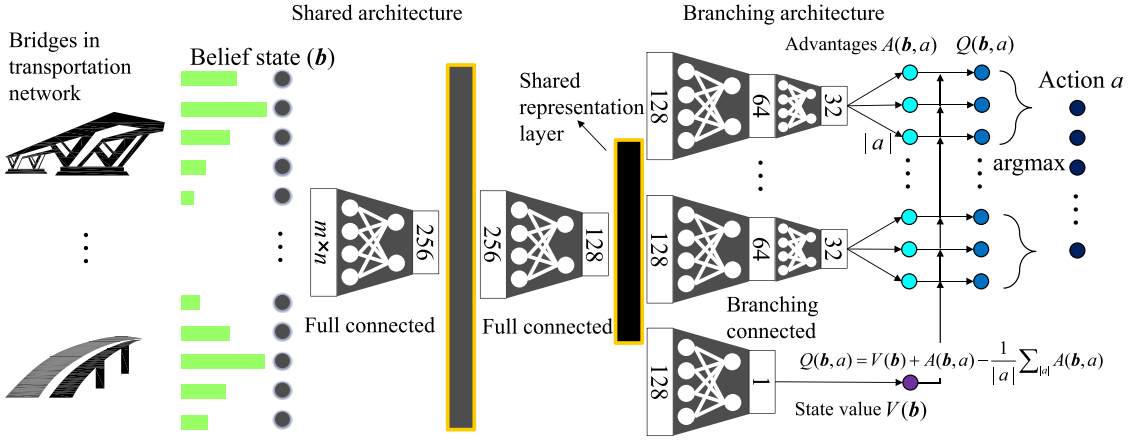

Fig. 4. Architecture of BDQN.

$$Q^j(\boldsymbol{b}_t, a_j; \theta_s, \theta_j) = V(\boldsymbol{b}_t) + \left[A(\boldsymbol{b}_t, a_j; \theta_s, \theta_j) - \frac{1}{|\mathbb{A}_j|} \sum_{d=1}^{|\mathbb{A}_j|} A(\boldsymbol{b}_t, a_d; \theta_s, \theta_j) \right] \tag{11}$$

In Eq. (11), the superscript of the letter j is designed for peculiarities of BDQN, which are consistence with the definitions in POMDPs (Section 2). Formally, for j^{th} bridge, the individual branch's output value Q^j at the current belief state $\boldsymbol{b}_t$ is expressed in terms of the common state value $V(\boldsymbol{b}_t)$ and the corresponding (state-dependent) sub-action advantage $A(\boldsymbol{b}_t, \boldsymbol{a}_t)$. $|\mathbb{A}_j|$ is the number of available maintenance actions in j^{th} bridge. θ_s denotes the parameters in the shared neural network, and θ_j refers to the parameters in a branching network.

It should note the slight difference in Q value between the BDQN and conventional DQN. The individual branch Q^j value is not consistent with the global state-action value Q in Eq. (7). In order to use global reward to update the loss function, branching Q value is summed, defined as:

$$Q(\boldsymbol{b}_t, a_j; \theta) = r(\boldsymbol{b}_t, a_t) + \frac{\gamma}{m} \sum_{j=1}^{m} \left(max_{a_{t+1}} Q^j(\boldsymbol{b}_{t+1}, a_{t+1}; \theta_s, \theta_j) \right) \tag{12}$$

In Eq. (12), the second item is divided by the number of branches m as the state value $V(\boldsymbol{b})$ is a duplicate considered. Moreover, each branch selects the action which can maximize its Q^j value and contributes $1/m$ to the total system Q value. For this, the loss function can be defined as the difference of each branching output, given as:

$$L(\theta) = \boldsymbol{E}_{\langle \boldsymbol{b}_t^j, a_t, r_t, \boldsymbol{b}_{t+1}^j \rangle \sim U(D)} \left\{ \frac{1}{m} \sum_{j=1}^{m} \left[r_t + \gamma max_{a_{t+1}} Q^- \left(\boldsymbol{b}_{t+1}, a_{t+1}; \theta_s^-, \theta_j^- \right) - Q^j(\boldsymbol{b}_t, a_j; \theta_s, \theta_j) \right]^2 \right\} \tag{13}$$

where the asynchronous updating method in double DQN is also utilized in BDQN. The parameter θ^- denotes the target networks which copy values to the main network ($\theta^- = \theta$) with an appropriate delay. Although BDQN has factorized the complex system action into the semi-independent branches action and the centralized neural network has the capability to coordinate the action across the branches, a single loss function (Eq. (13)) without considering differentiation among branches always leads to the final results far from the optimal solution. Relying on a single loss function could make it difficult to explicitly capture changes induced by multiple constraints in complex systems. For example, high maintenance costs, excessive CO_2 emissions, or traffic congestion can all result in negative feedback to the neural network. If all such feedback is simply fused into one loss function, the network may struggle to develop the ability to handle different situations with appropriate treatments. To combat this problem, hierarchical reward function incentives and adaptive learning rates are proposed to train the neural network.

3.2. Hierarchical multi-reward backpropagation method

The hierarchical multi-reward backpropagation BDQN (H-BDQN) is inspired by the biological learning behavior which commences with an unconditioned reflex for the simple task, then is followed by conditioned responses for complex assignments. In the early learning stages, the neural network is not fully converged. In this phase, if the neural network directly tackles the complex tasks, it will randomly attempt various action combinations. Once a combination brings positive feedback to the agent, the neural network parameters are adjusted so that this combination is favored in the corresponding belief state. However, this learning pattern difficulty distinguishes the actual effective actions in this combination. The experimental results demonstrate that the BDQN in this training method will fall into the conservative sub-optimal solution and lose the generalization ability to the new belief state. Therefore, the hierarchical multi-reward backpropagation approach is designed to take into account both individual-level and network-level feedback, which allows for precise control in complex systems. For transportation network management, the reward parameter, or maintenance fee in POMDPs can be divided into two parts, given as:

$$r(\boldsymbol{b}_t, a_t) = \sum_{j=1}^{m} r_j(\boldsymbol{b}_t^j, a_t^j) + r_E(\boldsymbol{b}_t, a_t) \tag{14}$$

where the first part (r_j) is the maintenance cost for individual bridges, and the second part (r_E) considers the effects of the action combination for the entire system. Obviously, if only the first reward component (r_j) is used to train the branching neural network, it would result in independent optimal policies for individual bridge management. This is because the loss value of one bridge's reward does not rely on the rewards of others. This learning process can be seen as analogous to a limb developing an instantaneous unconditioned reflex to external stimuli without central nervous control. While this may be the optimal response for a limb, it is not necessarily the best approach for the entire creature. Another reward parameter r_E incorporated with various sustainability-based constraints and mobility requirements is obtained after the agent interacts with the environment. Since the total reward r is the result of the synergy of many factors, the agent needs to sufficiently explore large discrete action spaces to understand complicated feedback mechanisms. After each branching network has developed the ability to deal with bridge maintenance, then, the central network can learn how to coordinate the conflicts between actions in different situations. Finally, the 'brain' of the neural network develops conditioned reflexes on the basis of the fundamental decision-making ability to accommodate the complicated environment.

The aforementioned contents are the theoretical backbone of the hierarchical multi-reward back-propagation method, and below we show the mathematical details. As Fig. 5 shows, there are two training mechanisms in H-BDQN, wherein the parameters of layers marked with black are updated in the training process. One uses the total reward (r in Eq. (14)) to train the entire network parameters (θ) with loss function Eq. (13). Another training mechanism uses the maintenance reward (r_j) to train the corresponding branching network parameters (θ_j) without updating the shared network (θ_s). The loss function $L(\theta_j)$ for branching is calculated based on the individual maintenance cost:

$$L(\theta_j) = \left(r_j(\boldsymbol{b}_t^j, a_t^j) + \gamma max_{a_{t+1}} Q_j^- \left(\boldsymbol{b}_{t+1}, a_{t+1}; \theta_s^-, \theta_j^- \right) - Q_j(\boldsymbol{b}_t, a_t; \theta_s, \theta_j) \right)^2 \tag{15}$$

It is worth noting that the magnitude of reward $r_j(\boldsymbol{b}_t^j, a_t^j)$ used to train the branching networks is inconsistent with the magnitude of $r(\boldsymbol{b}_t, a_t)$ for the entire networks. This magnitude discrepancy will induce distortion to the Q value produced by the neural network and will hardly converge. Hence, the normalization for parameters $r_j(\boldsymbol{b}_t^j, a_t^j)$ and $r(\boldsymbol{b}_t, a_t)$ is necessary.

Another key argument in the training process is the dynamic learning rate (η_j) which controls the branching networks (θ_j) updating speed. The dynamic learning rate plays a vital role in simulating a creature's learning process, and it also determines whether the proposed method can effectively improve the performance of neural networks. In the early

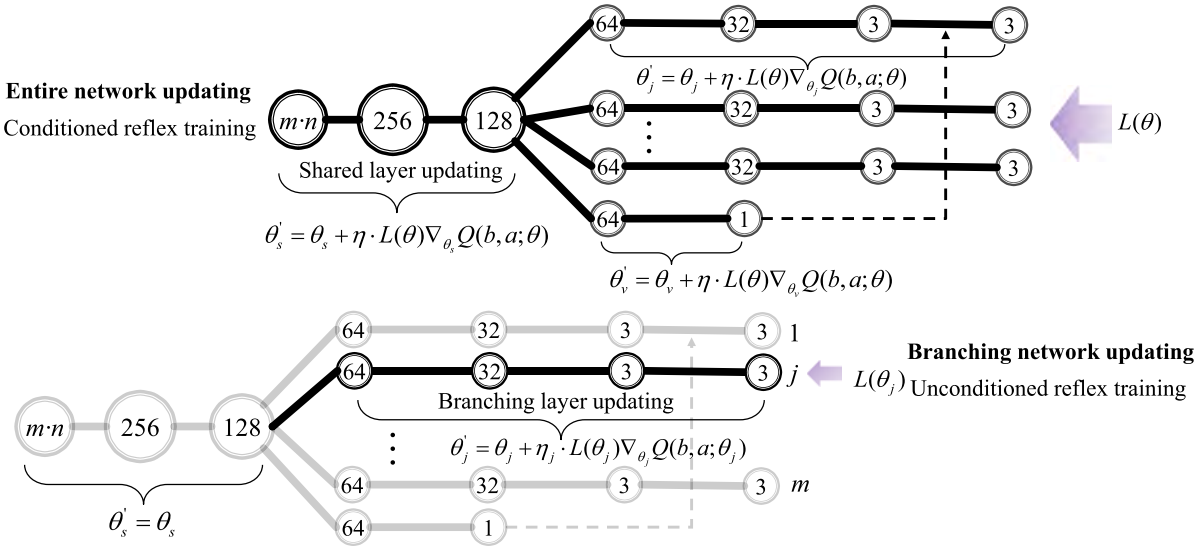

Fig. 5. Hierarchical multi-reward backpropagation method.

stage, a high branching learning rate (η_j) is necessary to ensure the priority of network learning in dealing with individual bridge management. After the branching networks develop their ability in dealing with single bridge maintenance, the learning rate (η_j) will gradually decrease to 0 through timing a discounting coefficient (ϕ) because the final objective is to manage the large transportation network. This process simulates biological forgetting which allows the branching to forget the established habit and accept new knowledge for complex assignments [61]. Apart from this, because of uncertainty and randomness in the environment, each branching network experiences a different learning process. In the practical training process, this difference may continually accumulate and significantly affect the final performance of branching networks. To address this problem, the concept of adaptive learning rate is introduced to encourage the poor-performance branching networks to have a higher learning capacity during the next iteration. The approach reflected in mathematical formulas is as follows:

As mentioned before, the maintenance cost $r_j\left(\boldsymbol{b}_t^j, a_t^j\right)$ and entire system feedback ($r(\boldsymbol{b}_t, a_t)$) are normalized as $0 \sim 1$. Through comparing the branch accumulative reward and system feedback in the life-cycle, each learning rate of the branching network is adjusted to:

$$\eta_k = 0.01 \frac{\sum_{t=1}^{L} r_j\left(\boldsymbol{b}_t^j, a_t^j\right)}{\sum_{t=1}^{L} r(\boldsymbol{b}_t, a_t)} + 0.99\eta_{k-1} \tag{16}$$

$$\eta_j = \eta_0 \eta_k \phi^k \tag{17}$$

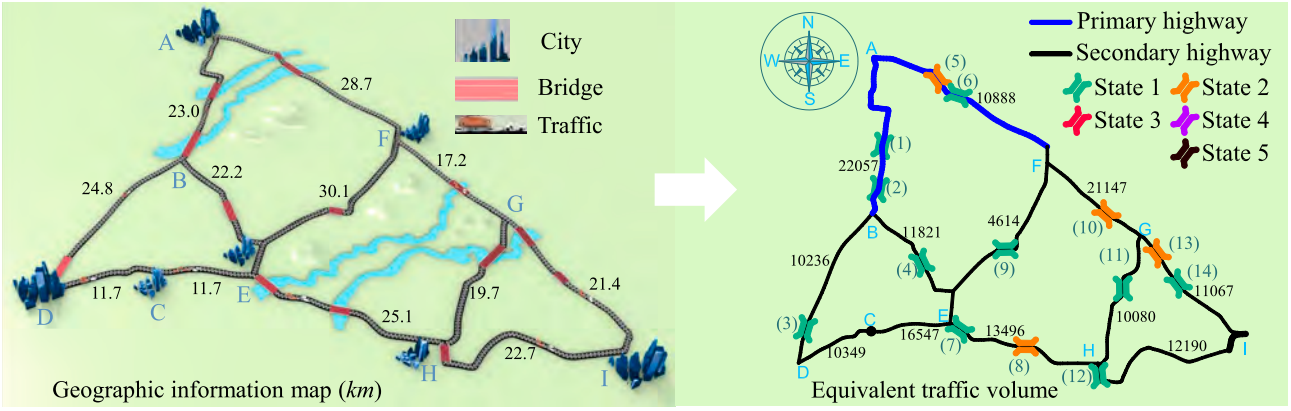

Fig. 6. Geographic information (left) and traffic information (right) for the investigated transportation network.

Table 1
Parameters of different vehicles [64].

Vehicle classification	Average fuel consumption (L/km)	Average CO_2 emissions (g/km)	Conversion ratio	Proportion
Motorcycle	3.5	80.5	1	6.00%
Tractor	6	157.5	4	1.66%
Car	20	252	1	22.38%
Medium bus	12	276	1	5.53%
Large bus	28	644	1.5	2.90%
Medium-duty truck	23	450	1.5	3.76%
Heavy-duty truck	33	867.9	3	3.36%
Extreme-duty truck	47	1236.1	4	7.25%
Extreme trailer	50	1315	4	2.60%

where η_0 is the initial learning rate ($\eta_0 = 1e^{-5}$), ϕ denotes the learning decreasing rate (0– 1) which determines the forgetting rate, k is the iterative number in training, and the η_k is the adaptive learning rate. In summary, the essential theory for large multi-component systems management is discussed in detail in this section. Other well-known techniques used in DRL, such as prioritized experience replay and the Adam optimizer, can be found in [62,63]. For numerical comparison and validation purposes, the comprehensive parameters of the neural network and training parameters are presented in Appendix A. The proposed algorithm is programmed in Python and detailed pseudocode is described in Algorithm 1 below for reproduction. The code can be found in https://github.com/Laili-engineering/Branch-deep-reinforcement-learning

database of bridge condition data in Hebei province.

The essential traffic data is presented in Table 1 and Table 2. The per-kilometer fuel consumption and carbon emissions for various vehicle types, as shown in Table 1 [64]. Based on Document No. 205 from the Ministry of Transport of China, vehicles of different sizes have distinct impacts on traffic flow, the traffic volume for different vehicle types needs to be converted into equivalent standard vehicles. Using the conversion ratio and proportion in Table 1, the actual number of different types of vehicles in the system can be calculated. For instance, the actual number of heavy-duty trucks is calculated as $4008 = 119353 \times 10.08\%/3$. To simplify the calculation, the proportion of different types of vehicles is divided by the conversion ratio. Hence, the number of heavy-duty trucks is given as $4008 = 119353 \times 3.36\%$. The

Algorithm 1: Hierarchical multi-reward backpropagation algorithm for BDQN

Input: Initial belief state $\boldsymbol{b}_0$;
Initialize replay buffer $\langle \boldsymbol{b}_t, a_t, r_t, r_t^j, \boldsymbol{b}_{t+1} \rangle \sim U(D)$;
Initialize BDQN main network and target network weights θ_0, θ_0^-;
Constructed the interacted environment (POMDPs and transportation network) ;
Output: Trained network weights θ;
for $episode$=1, N **do**
  Reset the initial belief state $\boldsymbol{b}_0$;
  for t=1, L **do**
    Calculate exploration rate ϵ which successively decreases in training process;
    With probability ϵ select a random action a_t^j;
    Otherwise select action $a_t^j = \text{argmax}_{a_t} Q_j(\boldsymbol{b}_t, a_t; \theta_s, \theta_j)$;
    Execute action in environment and collect reward $r_j(\boldsymbol{b}_t^j, a_t^j)$, r by Equation (14);
    Compute the belief state transition $\boldsymbol{b}_{t+1}$ through Equation (1);
    Store experience $\langle \boldsymbol{b}_t, a_t, r_t, r_t^j, \boldsymbol{b}_{t+1} \rangle$ in replay buffer $U(D)$;
    Sample minibatch from $U(D)$ based on prioritized experience;
    Calculate the system loss $L(\theta)$ by Equation (13) and branching loss $L(\theta_j)$ by Equation (15);
    if $mod(4) = 0$ **then**
      Update entire main network parameters:;
      $\theta^- = \theta + \eta L(\theta) \nabla_\theta Q(\boldsymbol{b}_t, a_t; \theta)$;
      Update branching network parameters with Equation (17):;
      $\theta_j^- = \theta_j + \eta_j L(\theta_j) \nabla_{\theta_j} Q(\boldsymbol{b}_t, a_t; \theta_j)$;
      Set $\theta^- = \theta$
    end
  end
end

4. Case study: real-world transportation system

4.1. Investigated transportation network

To demonstrate the applicability of the proposed framework in a large multi-component system and to prove superior performance over other methods, a case study based on a real-world transportation system in a city in northern China is analyzed. Fig. 6 displays a digital twin transportation system between the main city (Node A) and its adjacent cities. The geographic information system (GIS) from BigMap software is displayed in Fig. 6 (left), which highlights the city positions, bridge locations, river locations, and length (km) of traffic lanes. The component information in Fig. 6 (right) includes bridge conditions from the inspection database, road grade from GIS, and traffic volume from the Department of Transportation collection. It is important to note that the traffic flow in Fig. 6 is obtained after converting all different types of vehicles into equivalent standard vehicles. The necessary information on sustainable management is given as (i) the detailed traffic volume, which is refined to vehicles with different emissions, and (ii) the traffic flow values (after conversion to standard car) between each city in the transportation network are obtained from actual observational data collected by its stakeholders, as depicted in Table 2. The fundamental information of bridges is listed in Table 3.

Table 2
Parameters of traffic volume.

Traffic Line	Average vehicles (day)	Traffic Line	Average vehicles (day)
A ↔ C	6133	D ↔ E	1034
A ↔ D	18,503	D ↔ F	3014
A ↔ E	15,235	D ↔ H	1439
A ↔ F	2102	D ↔ I	12,219
A ↔ H	6214	E ↔ F	3985
A ↔ I	11,365	E ↔ H	2971
C ↔ D	1002	E ↔ I	3649
C ↔ E	612	F ↔ H	12,006
C ↔ F	1342	F ↔ I	8604
C ↔ H	1757	H ↔ I	3806
C ↔ I	2361		

Table 3
Parameters of bridges.

Bridge number	Span /width(m)	Structural style	Construction time	Condition
(1)	16/11	Simple supported bridge	1997	1
(2)	12/11	Multi-span simple supported bridge	1997	1
(3)	41.4/11	Multi-span hollow slab bridge	1997	1
(4)	36/10.5	Multi-span hollow slab bridge	1995	1
(5)	96/12	Multi-span hollow slab bridge	1997	2
(6)	40/13	Multi-span hollow slab bridge	2001	1
(7)	66.4/13.4	Prestressed concrete girder bridge	2020	1
(8)	51.5/11.2	Multi-span hollow slab bridge	2000	2
(9)	52.5/11.5	Multi-span hollow slab bridge	2009	1
(10)	106.8/12.4	Multi-span hollow slab bridge	1994	2
(11)	39/11.4	Multi-span hollow slab bridge	1998	1
(12)	484/12	Continuous T-girder bridge	1994	1
(13)	55/12	Multi-span hollow slab bridge	1994	2
(14)	40/11.4	Multi-span hollow slab bridge	1996	1

4.2. POMDPs-based bridge modeling

To construct the POMDP, the parameters of the 7-tuple $E = \langle \mathbb{S}, \mathbb{A}, \mathbb{O}, \boldsymbol{T}, \boldsymbol{\Theta}, \boldsymbol{R}, \gamma \rangle$ should be defined for each bridge [54], in conjunction with the bridge inspection database in Hebei province. Each bridge condition ($\mathbb{S}$) is classified into $1 \sim 5$ based on the Chinese standard (JTG/T H21–2011) which 1 denotes the intact state and 5 indicates failure. Accordingly, the total combination of the state is $5^m = 6.10e^9$, where $m = 14$ is the bridge number in the transportation network.

The observation value ($\mathbb{O} \in 1 \sim 5$) corresponds to the bridge state, obtained through bridge inspection. For the action space ($\mathbb{A}$), bridge inspection is necessary to be conducted at every time step since periodic inspections are essential according to Chinese standard (JTG/T H21–2011) [46]. Therefore, 'inspection' is not performed as a separate maintenance action. The prevalent maintenance actions ($\mathbb{A}$) could be approximately classified as 'Do nothing', 'Repair', and 'Rehabilitation' for each bridge. Similarly, the total action combinations are $3^{14} = 4.78e^6$.

The state transition matrix ($\boldsymbol{T}$) captures the stochastic nature of the deterioration process and it needs to be able to provide accurate predictions of future state. Although bridge deterioration is determined by multi-factors, such as material, age, traffic volume, span, bridge structural pattern, environmental factors, and so on. However, the excessive categorization of bridges will result in an insufficient number of bridges available for statistical analysis of degradation patterns related to a specific factor [65]. Through the comparative study of deterioration models [66], it is evident that the magnitudes of prediction errors based on homogeneous Markov chains are much higher compared to those in models using non-homogeneous state transition matrix based on structural style and material. For reinforced concrete (RC) bridges, the proportion of bridge condition rating is counted based on the data sourced by directly exporting records from the bridge inspection database, as shown in Fig. 7. Data post-processing is conducted to search the individual bridge records to filter the non-maintenance bridge during the period 2004–2021. As recommended by Wellalage [67], the representative state-transition matrix conforms to the unit-jump upper triangular form as follows:

$$T = \begin{pmatrix} p(1) & 1-p(1) & 0 & 0 & \dots & 0 \\ 0 & p(2) & 1-p(2) & 0 & \dots & 0 \\ 0 & \dots & p(i) & 1-p(i) & \dots & 0 \\ 0 & 0 & 0 & \dots & p(n-1) & 1-p(n-1) \\ 0 & 0 & 0 & 0 & \dots & 1 \end{pmatrix}_{n \times n} \tag{18}$$

n is bridge state number. The $p(i)$ refers to the probability that structure maintain the current state after servicing a year. If the current belief state is $\boldsymbol{b}_0$. Then state distribution $\boldsymbol{b}_t$ after t years is:

$$\boldsymbol{b}_t = \boldsymbol{b}_0 \cdot \boldsymbol{T}^t \tag{19}$$

Using the inspection database, the expected state $E(t)$ in time t is obtained. The $p(i)$ undetermined coefficient in state-transition matrix can be calculated by least square method:

$$min \sum_{t=1}^{L} |E(t) - \boldsymbol{b}_t|, \quad \text{subjected to } 0 \le p(i) \le 1, \quad \text{for i} = 1, 2, \dots, n-1 \tag{20}$$

L is the time considered (2004–2021). Based on the statistical data in Fig. 7, the state transition probability of RC bridge in the natural deterioration is fitted, given by Eq. (21).

$$\boldsymbol{T}_{RC} = \begin{pmatrix} 0.922 & 0.078 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.931 & 0.069 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.940 & 0.060 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.950 & 0.050 \\ 0.000 & 0.000 & 0.000 & 0.000 & 1.000 \end{pmatrix} \tag{21}$$

Because the bridge in poor condition (e.g., states 3 and 4) will be strengthened, this leads to a premature interruption of the natural deterioration processes. Therefore, the actual duration of poor condition for the structure is not fully observed and is only known to be as long as or longer than the observed duration, making it a right-censored observation. To complement the data deficiency, the remainder state transition probability is assumed based on the designed bridge lifespan. Similarly, the state transition matrix of the prestressed concrete bridge is

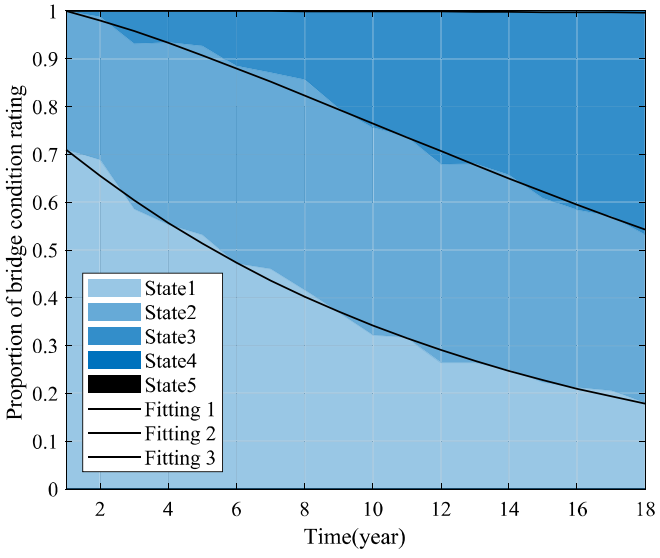

Fig. 7. Estimate state transition matrix with the inspected database by least square method.

obtained in a similar way, given as:

$$\boldsymbol{T}_{PC} = \begin{pmatrix} 0.906 & 0.094 & 0.000 & 0.000 & 0.000 \\ 0.000 & 0.957 & 0.043 & 0.000 & 0.000 \\ 0.000 & 0.000 & 0.950 & 0.050 & 0.000 \\ 0.000 & 0.000 & 0.000 & 0.960 & 0.040 \\ 0.000 & 0.000 & 0.000 & 0.000 & 1.000 \end{pmatrix} \tag{22}$$

Eq. (21) and (22) cover all bridge deterioration patterns in Table 3. For the 'Repair' maintenance, it should at least improve the performance of the structure, so the state transition probability should ensure a transition to a better state. The state transition matrix should be defined lower triangular matrix. The specific transition probabilities refer to assumptions from other papers [34,39]. assumed as [33]:

$$\boldsymbol{T}_{rp} = \begin{pmatrix} 1.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 1.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 0.050 & 0.950 & 0.000 & 0.000 & 0.000 \\ 0.050 & 0.100 & 0.850 & 0.000 & 0.000 \\ 0.000 & 0.050 & 0.150 & 0.800 & 0.000 \end{pmatrix} \tag{23}$$

Due to advanced techniques [68], the 'Repair' action could be conducted without interrupting the traffic. The effect of 'Rehabilitation' is to restore the structure to its initial state, with the following state transition matrix:

$$\boldsymbol{T}_{e} = \begin{pmatrix} 1.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 1.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 1.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 1.000 & 0.000 & 0.000 & 0.000 & 0.000 \\ 1.000 & 0.000 & 0.000 & 0.000 & 0.000 \end{pmatrix} \tag{24}$$

The adverse effect of 'Rehabilitation' is that the traffic line needs to be interrupted for an extended period, which is assumed to be 1 year in this study. The state-dependent observation matrix (Θ) is related to the periodical inspection in the bridge network. As expected, the bridge condition uncertainty decreases when inspections are taken, and the extent of reduction is proportional to the accuracy of the inspection. Assuming inspection can detect the correct state with probability (p), the state-dependent observation matrix is given in Eq. (25) [30]. The accuracy of inspection directly affects the ability to detect the state of the bridge, which in turn influences the choice of maintenance actions. Ideally, the accuracy is assumed as 1, maintenance actions are carried out based on the actual state of the bridge, eliminating any additional costs due to inspection errors. With observation errors increasing, the additional maintenance costs grow exponentially [29]. In this study, we do not focus on the sensitivity of inspected accuracy, but on the H-BDQN's superior performance in complex infrastructure management. The accuracy level of inspection is assumed to be at an acceptable level of $p = 0.9$.

$$\Theta = \begin{pmatrix} p & 1-P & 0.0 & 0.0 & 0.0 \\ \frac{1-p}{2} & P & \frac{1-p}{2} & 0.0 & 0.0 \\ 0.0 & \frac{1-p}{2} & P & \frac{1-p}{2} & 0.0 \\ 0.0 & 0.0 & \frac{1-\mathrm{p}}{2} & p & \frac{1-p}{2} \\ 0.0 & 0.0 & 0.0 & 1-p & p \end{pmatrix} \tag{25}$$

The reward matrix ($\boldsymbol{R}$) is influenced by a variety of factors, including the condition of the component, damage pattern, bridge pattern, span length, width, material, maintenance technology, and more. Even for components in the same condition, the same maintenance costs can vary greatly according to different standards (such as GB50500–2013 [69] and the USA Bridge Rehabilitation and Strengthening Manual PART-2 Cost Estimate). Therefore, it is not feasible to accurately determine the cost of a specific maintenance action, especially since the maintenance actions defined in this study are general for bridges and do not detail specific components or damage. For this reason, reference reward parameters ($r_j(s,a)$) are evaluated based on the USA standard. Given that all the bridges (Table 3) in the transportation system are made of concrete, it is assumed that the maintenance cost for each bridge is calculated by multiplying the unit price per square meter by the length and width of the damaged area, listed in Table 4. The negative value indicates each action is associated with a payment and the target of management is to minimize the cumulative cost in the life-cycle. To account for the effects of the condition of the bridge, it is assumed that maintenance costs increase exponentially at a rate of 1.02 with respect to the state. The discounting factor (γ) is assumed 0.99.

Table 4
Reward matrix table (unit: $k\$/m^2$).

Condition level	S1	S2	S3	S4	S5
Inspection + Do noting	−1	−1	−1	−1	−60
Inspection + Repair	−16	-16×1.02	-16×1.02^2	-16×1.02^3	−60
Inspection + Rehabilitation	−20	-20×1.02	-20×1.02^2	-20×1.02^3	−60

4.3. System-level sustainability cost and reward

The previous section introduces how the POMDPs are established for individual bridges, which has been widely studied [39,70]. However, the optimal solution at a single-component level cannot guarantee optimality at the system level because the interactions of different maintenance actions may reduce the sustainability indexes (e.g., social, and environmental) which are parameterized by mobility, and CO_2 emissions. In addition, as a result of traffic line dependence, the simplification that manually partitions the transportation into different regions will result in a large sub-optimal solution.

In this case study, the traffic is affected by two factors: maintenance and bridge condition. If the bridge is detected as state 3, the heavy truck, extreme duty truck, trailer, and tractor will be restricted from the passage. When the bridge is inspected as state 4 or executed 'Rehabilitation' action, the corresponding traffic line is restricted to all vehicles. The traffic control reshapes the interconnectivity of the network which induces detours for vehicles.

To reshape the traffic flow with lane interruption, Dijkstra's algorithm [72] is implemented to find the shortest paths between cities in the network. For instance, if the traffic lane B → D disruption, the traffic flow (18,503 in Table 2) between cities A ↔ D changes from A → B → D to A → B → E → C → D based on Dijkstra's algorithm. These detours induce two negative effects: On one side, the additional distance aggravates the CO_2 emissions from vehicles. On the other side, the excessive shutdown of bridges in the transportation network may increase the traffic volume larger than the designed maximum in the open roads. As mentioned in section 3.2 (Eq. (14)), the very important difference between single-component management and large-scale multi-component engineering systems is the multi-objective functions. System-level interdependence among components is reflected in the multi-objective function, with additional penalty mechanisms added to the environment at different system state configurations and action combinations.

The multi-objective function in this context comprises three reward functions: bridge safety as a risk factor, CO_2 emissions as a sustainability factor, and mobility as a traffic factor. However, these three reward functions represent the performance of the transportation system in three different units: bridge safety in $k\$$, CO_2 emissions in tons/year, and

Table 5
Normalized parameters for hierarchical reward function.

Baseline	Value (unit)	Weighting	Coefficient
Max CO2 emission	2.262e6 (tons/year)	Maintenance fee	$w_f = 0.7$
Min CO2 emission	1.131e6 (tons/year)	CO_2 emission	$w_c = 0.15$
		Mobility	$w_m = 0.15$
Highway grade	Service level (/day) [71]	Highway grade	Service level (/day)
Secondary highway	$V_4 = 22{,}000$	Primary highway	$V_3 = 24{,}000$
	$V_5 = 37{,}000$		$V_4 = 31{,}000$
			$V_5 = 35{,}000$
traffic paralysis	$r_t = -0.7$		

mobility in numbers/day. Both multi-objective optimization functions and the training of neural networks require the dimensionless processing of physical quantities to prevent one physical quantity from dominating and rendering other physical quantities ineffective due to large magnitudes. To uniformly consider different physical quantities, three reward parameters are normalized and combined with the weighting factor in Table 5, Eq. (14) is rewritten as:

$$r_t(b_t, a_t) = w_f \frac{1}{m} \sum_{j=1}^{m} r_j\left(b_t^j, a_t^j\right) + w_c r_c + w_m \frac{1}{l} \sum_{1} r_m \tag{26}$$

In Eq. (26), the first item (r_j) is the normalized maintenance fee for each bridge, and m is the bridge number in the transportation system. w is the weighting factor. The second item (r_c) is the normalized carbon emissions of whole transport vehicles. The third item (r_m) normalizes the traffic volume of the entire transportation network. l is the number of traffic lines.

$$r_j\left(b_t^j, a_t^j\right) = -\sum_{j=1}^{m} \frac{r_j\left(b_t^j, a_t^j\right)}{r_{max}^j} \tag{27}$$

$$r_c = -\frac{CO_2 - CO_{2min}}{CO_{2max} - CO_{2min}} \tag{28}$$

$$r_{m,p} = \begin{cases} 0 & for\ V \le V_3 \\ -1/3 & for\ V_3 \le V \le V_4 \\ -2/3 & for\ V_4 \le V \le V_5 \\ -1 & for\ V_5 \le V \end{cases} \tag{29}$$

$$r_{m,s} = \begin{cases} 0 & for\ V \le V_4 \\ -1/2 & for\ V_4 \le V \le V_5 \\ -1 & for\ V_5 \le V \end{cases} \tag{30}$$

For maintenance fee normalization, the minimum maintenance cost is 0 (refers to 'Do nothing'). The dimensionless process uses the maintenance fee dividing the maximum action cost, calculated by Eq. (27). Since each bridge has different widths and lengths, the maximum maintenance cost (r_{max}^j) is determined by multiplying the unit cost of 'Rehabilitation' actions with the bridge's span and width.

In the case of carbon emissions, given that vehicles travelling the road network will inevitably produce CO_2, there will be a minimum and maximum value for these emissions. Hence, to dimensionally normalize the carbon emission values, the linear normalization method is employed. The low-bound and high-bound calculations are as follows. When the whole of the bridges in the transportation system are in good condition (state 1 or state 2), vehicles in the system do not need to detour and can take the shortest path to reach their destination. Based on the distance (Fig. 6) traveled by vehicles (Table 2), the minimum carbon dioxide emissions (Table 5) can be calculated. Similarly, if all vehicles take the longest distance detour to reach their destination, the maximum carbon dioxide emissions are obtained. Using the maximum value as the upper bound ($CO_{2\ max}$) and the minimum value as the bottom bound ($CO_{2\ min}$), the carbon emissions of transport vehicles can be normalized in any situation, given by Eq. (28).

Regarding mobility, according to the highway design standards [73], the service level of a highway is reflected in its traffic flow and congestion. Each service level corresponds to a maximum daily traffic volume. This traffic volume is determined by factors such as driving speed, intersections, and the proportion of different types of vehicles. It can be calculated based on the JTG D20–2006 standard [71]. The service capacities of primary and secondary highways are listed in Table 5. The subscript of V_i in Eqs. (29) and (30) means the grade of highway service. For instance, the V_3 in primary highway is the threshold of service grade which means the traffic volume <24,000 per day satisfies level-3 requirements. The normalized reward of mobility (r_m) in each traffic line (l) is obtained based on different service grades, given in Eq. (29) for the primary highway and Eq. (30) for the secondary highway.

Finally, the weighting factors (w) are determined by the Analytic Hierarchy Process (AHP) method [74]. It includes three major steps, decomposition, comparative judgment, and synthesis of priorities. Decomposition involves breaking down the objective function into its constituent parts which are maintenance, carbon emissions, and mobility. Comparative judgment involves assessing the relative importance of three parts. Herein, the important ratio between maintenance and carbon emissions is 5, between maintenance and mobility is 5, and between carbon emissions and mobility is 1. Combining the AHP matrix [74], the synthesis of priorities for weighting factors are given in Table 5. It should be noted that the weighting factor will determine the final value of multiple objective function (26) and significantly affect the optimal management policy. The sensitivity of weighting factors will be discussed in Appendix B.

Using the weight factor, the multiple reward parameters are defined as Eq. (26) and update the entire neural networks through Eq. (13). The reward parameters should also highlight the particular situation that the excessive traffic lines are interrupted which paralyzes the transportation system. For instance, if bridges in A ↔ B and A ↔ F simultaneously deteriorate to state 3, the truck is impassable due to the risk control. Since the transportation network is severely disrupted, a punishment $r_t = -0.7$ is set to avoid exploring action combinations that might lead to traffic paralysis during training. After establishing the environment, algorithm 1 is implemented to train the neural network. In the remainder of this study, the outlined contributions are introduced in detail along with the large-scale multi-component control problem and a comprehensive discussion of results.

5. Results and discussion

5.1. Comparative study

In this section, the improvement of the proposed framework on the above challenging control problem with high action dimensionality and complexity is demonstrated. Four maintenance policies are compared, of which the first three are developed by neural networks using different algorithms: DDPG with Wolpertinger architecture (DDPG-W), BDQN, and H-BDQN. The fourth policy is a routine maintenance strategy designed to maintain the bridge condition without interrupting traffic. The cumulative reward in the life-cycle (100 years) is regarded as a

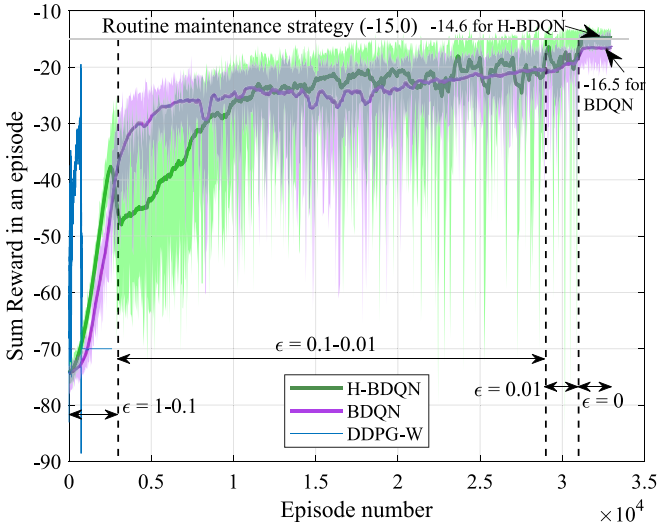

Fig. 8. Cumulative reward during the training process.

criterion to evaluate the improvement of the policy during the training process. The training details are as follows: an episode is the transportation network experiencing 100-time steps. For DDPG-W, the updating happens in each time step. For BDQN, every 2-time steps, and For H-BDQN, every 4-time steps. As controlling the random seed method is not applicable for TensorFlow GPU training, to mitigate the impact of randomness on the final results, all models were trained 100 times, with the best result subsequently selected. The performances are summarized in Fig. 8. This study first compares the performance of the BDQN against DDPG-W on the large-scale multi-component transportation system control problem. The better cumulative reward curve demonstrates that BDQN outperforms the DDPG-W method. To clearly exhibit the behavior developed in different learning methods, the video of the entire training process can be viewed at https://www.bilibili.com/. It can be readily noted that the inconsistency between the continuous action representation and actual discrete action spaces makes DDPG-W challenging to converge. The superior performance of BDQN in comparison to DDPG-W also verifies the effectiveness of the shared network module in coordinating individual bridge maintenance and entire transportation system management.

Additionally, the cumulative reward discrepancy between the H-BDQN and BDQN models indicates their different learning capabilities during the training process. Fig. 8 displays the cumulative reward improvement during the training process and 2000 Monte Carlo simulations after training. It can be concluded from Fig. 8 that the H-BDQN model enables the efficient improvement of policy, as the mean slope of the green line is larger than that of the violet line, and the final policy has the best performance. In fact, these two neural networks develop distinct transportation network management strategies during the training process. Two representative policy realizations are visualized in Figs. 9 and 10, highlighting different maintenance behaviors. To account for random factors, each maintenance policy is executed with 2000 Monte Carlo simulations, and the box plot of cost is shown in Fig. 11.

Continuing the discussion in relation to the difference between H-BDQN, BDQN, and routine maintenance strategy, these three maintenance policies are further contrasted. In the studied case, 14 bridge maintenance strategies are displayed with 'Do nothing', 'Repair' (△), and 'Rehabilitation' (⊡) actions in Fig. 9 and Fig. 10. The blue line depicts the actual state development ($\boldsymbol{b}_t \rightarrow \boldsymbol{b}_{t+1}$) in the life-cycle. Due to the uncertainty in inspection, the observed state ($\boldsymbol{o}_t \rightarrow \boldsymbol{o}_{t+1}$) is slightly different from the actual state (90% accuracy) which is described by the green line. Two maintenance actions ('Repair' and 'Rehabilitation') improve the bridge's performance and avoid failure. The 'Repair' action can enhance the bridge condition without affecting the transportation system, but the action has low cost-performance. On the other hand, the 'Rehabilitation' action has high cost-performance in maintenance, but it may induce a detour, additional CO_2 emissions, and traffic congestion. The purpose of transportation network management is to implement appropriate maintenance actions to maximize the value of the multi-objective function (27). In the following contents, the maintenance policies will be evaluated in two aspects, intuitive reasonability and value function (27), based on Figs. 9, 10, and Fig. 11.

Under the BDQN method, considering the randomness of neural network training, we train 100 groups of neural networks separately and take the group with the best performance. However, even with this approach, the maintenance policy still falls into sub-optimal solutions.

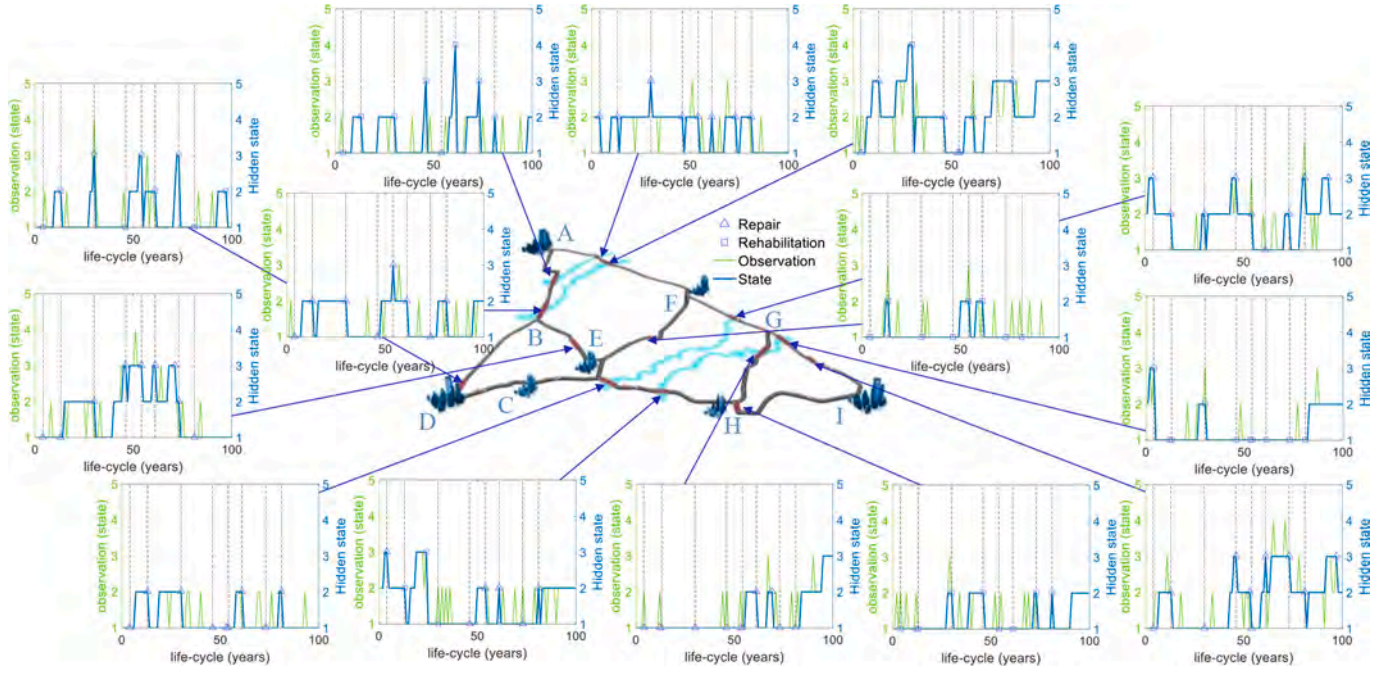

Fig. 9. Transportation networks management by BDQN.

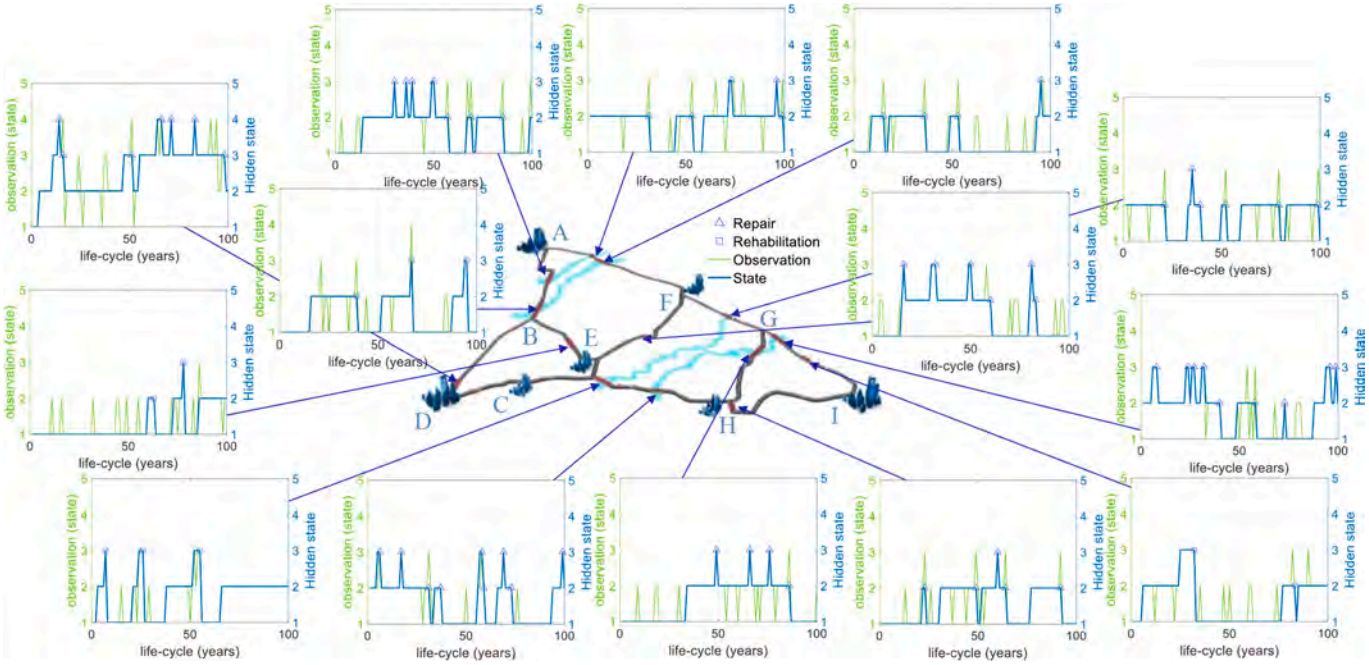

Fig. 10. Transportation networks management by H-BDQN.

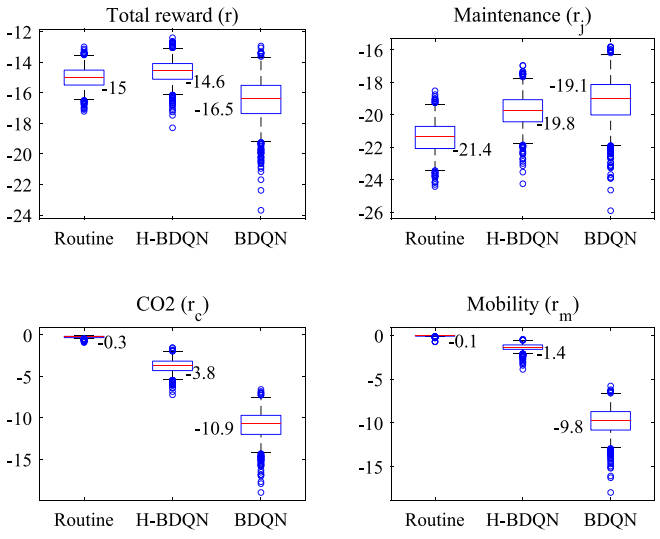

Fig. 11. Comparison of cost among different maintenance policies.

The most straightforward criterion for judging the reasonableness of the management policies is whether the maintenance action is taken when the bridge is in poor condition. However, as shown in Fig. 9, the dashed line indicates that a blind maintenance action is adopted for all bridges when parts of the bridge degrade to poor conditions. This blindness leads to the implementation of maintenance actions on intact bridges and induces additional maintenance costs. Furthermore, the policy developed by BDQN does not fully recognize the environmental constraints that can induce traffic paralysis. This is evident in Fig. 9, where the maintenance actions are shown in the third dashed line. The unreasonable action combinations are implemented, which simultaneously adopted 'Rehabilitation' in bridges (1) and (6), interrupting the traffic flow at traffic lanes A ↔ B and A ↔ F. The entire transportation system is paralyzed. Lastly, as the worst life-cycle performance (−16.5) shown in Fig. 11, the sub-optimal solution dramatically increases the cost of CO_2 emissions (−10.9) and mobility (−9.8).

Compared with BDQN, the routine maintenance strategy attempts to avoid situations that affect the traffic flows in the transportation system. Therefore, the 'Rehabilitation' and state over 3 are not allowed in this strategy. The 'Repair' action is implemented when inspecting the corresponding bridge state deteriorates to state 3. Nevertheless, the routine maintenance strategy is excessively conservative. As shown in Fig. 11, frequent and uneconomical 'Repair' actions can lead to the highest cost (−21.4) among the three maintenance strategies.

The metric to evaluate a good DRL method is to testify whether the trained policy can surpass the general level of experts, and in turn, guide the management of the transportation network. As the information in Fig. 10, all maintenance actions ('Repair' and 'Rehabilitation') are implemented when the bridge's state is observed to be in poor condition (state 3). Hence, the policy from H-BDQN at least satisfies optimal maintenance requirements for the bridge individually. For system-level management, H-BDQN obtains the best performance (−14.6) among different models in a defined environment according to Fig. 11. The maintenance policy finds the trade-off to adopt the 'Repair' action and 'Rehabilitation' action without paralyzing the traffic system and saving the budget as much as possible, as shown in Fig. 10. Similar to other DRL methods [75] that introduce innovative problem-solving approaches or new knowledge, H-BDQN also enhances management capabilities through interaction with the environment, providing guidance for managers in complex system management. Since the bridge's condition will affect the passage of lanes, traffic lanes with multiple bridges can be more frequently interrupted compared with traffic lanes with a single bridge. Therefore, a relatively higher budget needs to be spent if the agent maintains the passage of lanes with multiple bridges. Obviously, the agent discovers this property and utilizes it to minimize the cost with the required performance. The vulnerable traffic lanes with multiple bridges, such as A ↔ B, E ↔ H, G ↔ I, and A ↔ F, are marked as the candidates which may adopt the'Rehabilitation' to minimize the maintenance fee and allow temporary interruption. Simultaneously, to maintain the entire transportation system operation, the agent needs to reserve one of the lanes, A ↔ B or A ↔ F, with the 'Repair' action.

Another important notable performance of H-BDQN is that it trades off millions of situations and discovers the most economical maintenance mode, which keeps the passage of traffic lane A ↔ F. Since no bridge is located at the traffic lane D ↔ E and no city in B point, to

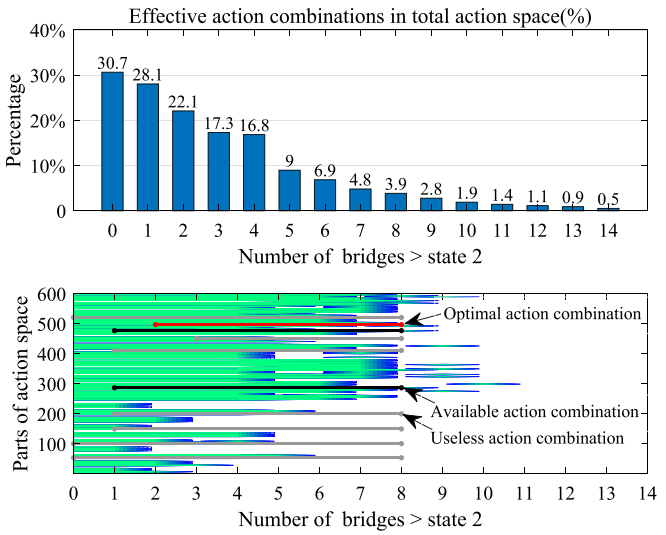

Fig. 12. Solution space in different bridge deteriorated conditions.

further cut cost, the lane B ↔ D and B ↔ E allow temporary interruption. The agent maintenance behavior provides a new way of thinking for managers, compared with the conservative routine maintenance strategy.

5.2. Decision-making network learning mechanism

Herein, a detailed explanation is provided for the reasons behind the differences in maintenance policies resulting from various training methods. When the bridges in the transportation network deteriorate to state 3, or when the 'Rehabilitation' action is taken, it can affect vehicle passage, potentially leading to traffic paralysis and punishments. To gain insight into how the condition of the bridges affects the maintenance action combinations, assuming the bridges in the transportation network deteriorate to state 3 in the order of bridge serial number in Fig. 6. The upper half of Fig. 12 displays how the available action combinations without paralyzing traffic decrease as the bridges deteriorate. In the lower half of Fig. 12, the available action combinations that do not violate the environmental constraints are visualized using blue contour lines. The low percentage of available action combinations occupied in total action spaces indicates large action spaces are worthless to explore, as the gray lines (useless action combinations) marked in Fig. 12. In the early stages of training, as shown in the video https://www.bilibili.com/, the large stochastic exploration factor ensures that parts of the available action combinations can be discovered, as shown by the black lines in Fig. 12. In the medium-term training stage, the goal of training is to find the optimal maintenance action (represented by the red line) to substitute the sub-optimal one in any state. Therefore, how to filter or avoid the exploration in useless action space determines the effectiveness of the training method.

For conventional BDQN with a single feedback mechanism, information loss occurs during the fusion process of environmental feedback rewards. This oversimplified reward feedback cannot offer insight into the influence of different actions. For instance, the neural network will learn from the feedback that the 'Repair' action will always improve transportation networks, but the 'Rehabilitation' action affects the performance randomly because substantial combinations of 'Rehabilitation' actions will paralyze traffic. However, the information that 'Rehabilitation' action can improve the corresponding bridge condition is lost during training. Random exploration with unfiltered action combinations may result innumerous attempts that yield negative feedback, causing the neural networks to maintain the current policy. This training behavior is reflected in Fig. 8, the unchanged policy leads to a low reward fluctuation. In the late training, the generalization ability of the neural network decreased and finally fall into the sub-optimal solution.

For the H-BDQN, the reward feedback from the environment is disassembled into two parts: the maintenance cost for individual bridges and the global reward of the transportation network. Through multi-reward Eq. (16), the branching network updates parameters to accommodate the safety management for the individual bridges in the early training phase. This targeted adjusting facilitates the neural networks to find maintenance action combinations for bridges in poor conditions. Specifically, the useless action combinations are filtered. Afterward, with the forgetting coefficient increasing during the medium term, the learning rate for branching networks successively decreases, and simultaneously, entire network parameters are updated by the global loss function. This synergetic multi-reward training mode preserves a certain extent autonomy of the branching, which allows the branching network to attempt different maintenance actions for the corresponding bridge. Then, the unreasonable combinations of maintenance may induce punishment from the environment, which is passed through neural networks through global loss function. The negative feedback helps the agent to recognize the constraints and continually coordinate the maintenance policy in each branching. Once the virtuous circle is developed, the competitively self-adjusting capacity motivates the agent to sufficiently explore the environment and converge to an efficient near-optimal solution. This learning feature is demonstrated in Fig. 8 which has more drastic fluctuation and a larger convergence rate.

5.3. Discussion

This study has two main contributions. First, we embed various sustainability metrics in the transportation network management using POMDPs, including structural safety, carbon emissions, and mobility considerations. Unlike common management practices that may simplify the effects among different actions or just manage bridges individually, a DRL method named H-BDQN is developed to consider the effects of every action combination. The second contribution is the development of a novel learning method that simulates biological learning behavior, from unconditioned reflex to conditioned reflex. The results show that the network trained by this method performs better in complex system control problems than the traditional DBQN, by searching global optimal solutions better and avoiding falling into sub-optimal policy regions. These two properties contribute to the proposed framework performing adequately well in network-level multi-component systems management.

However, for network-level management, when the number of bridges is in the order of hundreds or thousands, H-BDQN also faces difficulty in achieving good near-optimal results. For managing transportation networks of this scale, it becomes crucial to leverage known road network information to simplify the model. Some effective methods include employing graph neural networks to quantify the connectivity and length of traffic lanes. Another approach involves using a dataset to pre-train the neural network. In fact, a large number of state-action combinations that could lead to the paralysis of the entire road network can be utilized to compile such a dataset. Further in-depth research is needed in this area.

6. Conclusions

A versatile deep reinforcement learning model for tackling the important problem of scheduling comprehensive maintenance in large-scale multi-component engineering systems is developed in this paper. The real-world transportation network is quantified as the environment in deep reinforcement learning, the connections of roads are established by Dijkstra's algorithm, and maintenance decision-making for bridges is formulated as POMDPs. Additionally, hierarchical reward functions were designed to meet sustainable requirements (e.g., economic, social, and environmental), which facilitate stakeholder management.

However, the lossless information quantification induces the curse of dimensionality in state and action expression. To address this issue, using the Branching Dueling *Q*-Networks, the large complex action space is factorized, making the nodes of the output layer grow linearly with the number of bridges. Preventing the solution from falling into local optima is another problem of complexity in large-scale system control. This study designs a hierarchical multi-reward backpropagation method for optimal policy training to facilitate the neural networks to accommodate decision-making in a complex environment gradually. The training process simulates the biological learning behavior, from simple disassembling tasks to integrated complex assignments. Through comparing results between state-of-art methods in large discrete action spaces, the H-BDQN trained with the proposed method displays strengths related to stability in training, exploration of optimal policies, and convergence speed.

The study yielded the following insights into maintenance behaviors: (a) Routine maintenance strategy, only adopting 'Repair' actions, can minimize negative effects on traffic flows (e.g., detours and mobility), but it is not an economical maintenance strategy for bridge performance. A good maintenance strategy should consider the vulnerable traffic lane with multiple bridges, the passage of the transportation network, and the utility of different maintenance actions, as inspired by the policy developed by the proposed method. (b) Based on the maintenance policies in the study case, a reasonable maintenance strategy for transportation system management should include the following steps: 1) calculating the maintenance cost of each traffic lane without interruption; 2) ranking the high-cost lanes and implementing 'rehabilitation' actions with temporary interruption; 3) identifying available combinations of traffic lane interruption that will not significantly impact the transportation network; and 4) determining the most economical and optimized scheme.

CRediT authorship contribution statement

Li Lai: Writing – original draft, Visualization, Software, Methodology, Formal analysis, Data curation, Conceptualization. **You Dong:** Supervision, Project administration, Funding acquisition, Conceptualization. **Charalampos P. Andriotis:** Writing – review & editing, Supervision, Funding acquisition. **Aijun Wang:** Validation, Software, Methodology, Formal analysis. **Xiaoming Lei:** Writing – review & editing, Resources, Methodology, Investigation, Funding acquisition, Conceptualization.

Declaration of competing interest

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

Data availability

Data will be made available on request.

Acknowledgements

This study has been supported by the Research Grants Council of Hong Kong (PolyU 15225722), the Innovation and Technology Commission of Hong Kong SAR Government to the Hong Kong Branch of National Engineering Research Center on Rail Transit Electrification and Automation (K-BBY1), the Research Institute for Sustainable Urban Development, the Hong Kong Polytechnic University (PolyU 1-BBWM), and Centrally Funded Postdoctoral Fellowship Scheme (PolyU 1-YXB5), the TU Delft AI Labs program. The support is gratefully acknowledged. The opinions and conclusions presented in this study are those of the authors and do not necessarily reflect the views of the sponsoring organizations.

Appendix A. Appendix

The network implementations in this work are fully described in this appendix, including input, hidden layer, output, activation function, connection, and training process. The entire program (algorithm 1) is coded by Python libraries of TensorFlow. H-BDQN does not employ the fully connected structure and one gradient updating entire network parameters. In the input layer, every five nodes represent a bridge condition from state 1 to state 5. The input vector (state number $\times$ bridge number) is real numbers describing a probability distribution over all bridge damage states, shown in Fig. 4. The following is a shared part with two fully connected ReLU layers, 70×256 and 256×128, respectively. Then, the branching part adopts a relatively independent structure in which the internal of the branching network is fully connected ($128 \times 64 \times 3$) but each branching network is isolated. To reduce the repercussion of isolation and simplify the learning process, the dueling architectures are employed to express the state-action value (Q) into state value (V) and the state-dependent action advantage (A). As Fig. 4 shows, the final branching is a fully connected layer (128×1) to estimate state values. For bootstrapping-based algorithms, this separation increases the stability of the optimization and learning rate. The updating of action advantages can change as fast as the mean, instead of having to compensate for any change to the optimal action's advantage. Finally, each action advantage is aggregated with state value to produce the Q function. The output of each branching network is the state-action value Q_j of the current bridge for the entire system

A number of training techniques are utilized to enhance stability, generalization ability, exploration ability, and to avoid overfitting. Parametric updates are executed through mini-batch gradient descent with a size of 32, as it facilitates training speed and simultaneously maintains satisfactory convergence properties. The batch sampling is based on the prioritized experience replay [63] which increases the sampling probability of experience that has a larger difference between the target network and the main network. Compared with the uniform experience replay, prioritizing experience can sample important experiences more frequently, and learn more efficiently. The Adam optimizer is used for the entire network and branching networks. For the entire network, the initial learning rate is 10^{-5} and keeps constant in training progress. For branching networks, the synergetic multi-reward learning method in Eq. (16) is utilized to dynamically adjust the learning rate. The delay factor (φ) is set as 0.99997 to accommodate the 30,000-episode training processes. The parametric updates of target networks are executed every 4 action steps, whereas an exploration rate starting from 100% linearly decreases to 10% in the first 3000 episodes, and continually decreases to 1% at the end of training. A self-evaluation mechanism is incorporated into the network training process. Specifically, after every 5000 training iterations, the network's performance is evaluated using 2000 Monte Carlo simulations. If there is an improvement of $>5\%$ in performance compared to the previous neural network, the current network is preserved. If not, the network is discarded and the parameters from the preceding network are reinstated.

Appendix B. Appendix

Two types of objective functions are set in this work, the first type is related to the individual bridge, such as the maintenance fee (r_f), while the second type is associated with the entire transportation system, such as CO_2 emissions (r_c) and mobility (r_m). The weighting factors (w_f, w_c, and w_m) in the multi-objective function determine which type of objective is prioritized in the management policy. In this transportation system management, 'Do nothing' action and two maintenance actions (e.g., 'Repair' and 'Rehabilitation') are needed to be selected by the policy. 'Repair' induces a high maintenance fee and has no impact on effects on CO_2 emissions and mobility. 'Rehabilitation' is a cost-effective action but may induce CO_2 emissions and congestion. As the bridge deterioration also influences traffic flow, maintenance actions are inevitable. The optimal policy is a trade-off to use 'Repair' and 'Rehabilitation' at the appropriate bridge and time

Therefore, the weighting factor will directly decide the optimal maintenance policy. To gain insight into the influence of these weighting factors, an additional case is provided. A new weighting factor ($w_f = 0.5$, $w_c = 0.25$, and $w_m = 0.25$) is assumed. Through H-BDQN, the optimal maintenance policy of the new weighting factor is shown in Fig. 13. The corresponding objective function value (r, r_f, r_c, and r_m) is evaluated by 2000 Monte Carlo simulations, given in Fig. 14. The results indicate that the interruption of traffic lanes B⟷E and G⟷H will not significantly contribute to an increase in CO2 emissions and traffic congestion.

Herein, the optimal maintenance policies with different weighting factors are compared. H-BDQN with weighting factor ($w_f = 0.7$, $w_c = 0.15$, and $w_m = 0.15$) are defined as model 1, H-BDQN with weighting factor ($w_f = 0.5$, $w_c = 0.25$, and $w_m = 0.25$) are defined as model 2. Additionally, we assume another model, denoted as Model 3, with a special weighting factor ($w_f = 0$, $w_c = 0.5$, $w_m = 0.5$). Although we do not train H-BDQN of model 3, it can still infer the optimal maintenance policy. As maintenance cost is not considered ($w_f = 0$), the optimal maintenance policy is to prevent traffic interruption. The most effective policy would be to use the 'Repair' action to improve the state of the bridge and prevent it from deteriorating to state 3. In this situation, the optimal maintenance policy is identical with the routine maintenance strategy.

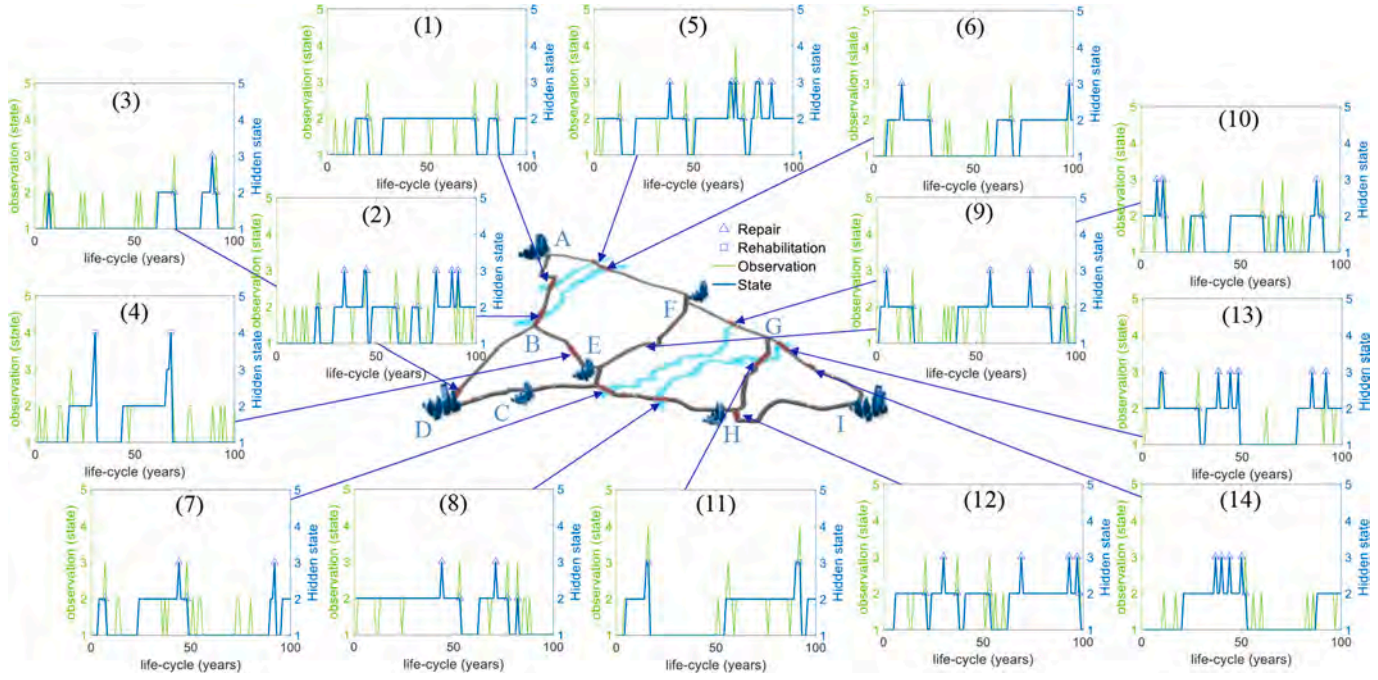

Fig. 13 Transportation networks management by H-BDQN with weighting factor ($w_f = 0.5$, $w_c = 0.25$, $w_m = 0.25$).

The right side of Fig. 14 shows that under certain weighting factors, all three models reach their optimal solutions. The left side of Fig. 14 displays the maintenance behavior of different policies. For the weighting factor ($w_f = 0.7$, $w_c = 0.15$, and $w_m = 0.15$), H-BDQN in model 1 ensures the transportation network's operation while saving maintenance costs ($r_f = -19.8$) as much as possible. With the weighting factor ($w_f = 0.5$, $w_c = 0.25$, $w_m = 0.25$), H-BDQN in model 2 shift policy to adopt cost-effective maintenance actions while minimizing the impact on traffic flow ($r_c = -1$ and $r_m = -0.3$) as much as possible. In summary, H-BDQN is an effective method to find the optimal maintenance policy and can sensitively adjust the policy with the weighting factor changing.

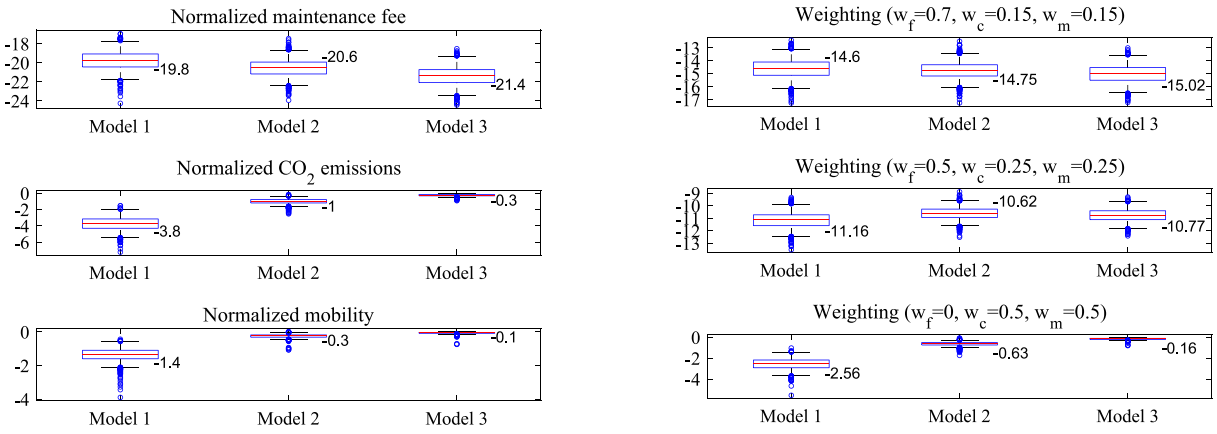

Fig. 14 Monte Carlo results of various maintenance policies under different weighting factors. The figure illustrates how the optimal maintenance policy changes with the variation in weighting factors. It provides a visual representation of the effectiveness of different maintenance policies under different scenarios, demonstrating the flexibility and adaptability of the H-BDQN model in finding the optimal maintenance policy.

References

[1] R. Victor, G. Baskir, J. Bennett, J. Camp, R. Capka, S. Curtis, G. Davids, L. Frevert, H. Hatch, A. Herrmann, Report Card for America's Infrastructure, American Society of Civil Engineers, 2013. https://infrastructurereportcard.org/cat-item/bridges-infrastructure.

[2] A.O. Acheampong, J. Dzator, M. Dzator, R. Salim, Unveiling the effect of transport infrastructure and technological innovation on economic growth, energy consumption and CO2 emissions, Technol. Forecast. Soc. Chang. 182 (2022) 121843, https://doi.org/10.1016/j.techfore.2022.121843.

[3] P. Bocchini, D.M. Frangopol, A probabilistic computational framework for bridge network optimal maintenance scheduling, Reliab. Eng. Syst. Saf. 96 (2011) 332–349, https://doi.org/10.1016/j.ress.2010.09.001.

[4] X. Zhao, X. Ma, B. Chen, Y. Shang, M. Song, Challenges toward carbon neutrality in China: strategies and countermeasures, Resour. Conserv. Recycl. 176 (2022) 105959, https://doi.org/10.1016/j.resconrec.2021.105959.

[5] X. Lei, Y. Dong, D.M. Frangopol, Sustainable life-cycle maintenance policymaking for network-level deteriorating bridges with a convolutional autoencoder–structured reinforcement learning agent, J. Bridg. Eng. 28 (2023) 04023063, https://doi.org/10.1061/jbenf2.Beeng-6159.

[6] H. Zhang, G.A. Keoleian, M.D. Lepech, Network-level pavement asset management system integrated with life-cycle analysis and life-cycle optimization, J. Infrastruct. Syst. 19 (2013) 99–107, https://doi.org/10.1061/(asce)is.1943-555x.0000093.

[7] D.M. Frangopol, M. Liu, Maintenance and management of civil infrastructure based on condition, safety, optimization, and life-cycle cost, Struct. Infrastruct. Eng. 3 (2007) 29–41, https://doi.org/10.1080/15732470500253164.

[8] X.M. Lei, L.M. Sun, Y. Xia, Lost data reconstruction for structural health monitoring using deep convolutional generative adversarial networks, Struct. Health Monitor.-Int. J. 20 (2021) 2069–2087, https://doi.org/10.1177/1475921720959226.

[9] M. Xu, M. Ouyang, L. Hong, Z.J. Mao, X.L. Xu, Resilience-driven repair sequencing decision under uncertainty for critical infrastructure systems, Reliab. Eng. Syst. Saf. 221 (2022) 108378, https://doi.org/10.1016/j.ress.2022.108378.

[10] X.M. Lei, D.M. Siringoringo, Y. Dong, Z. Sun, Interpretable machine learning methods for clarification of load-displacement effects on cable-stayed bridge, Measurement 220 (2023), https://doi.org/10.1016/j.measurement.2023.113390.

[11] C. Wu, P. Wu, J. Wang, R. Jiang, M. Chen, X. Wang, Critical review of data-driven decision-making in bridge operation and maintenance, Struct. Infrastruct. Eng. (2020) 1–24, https://doi.org/10.1080/15732479.2020.1833946.

[12] E.M. Abdelkader, O. Moselhi, M. Marzouk, T. Zayed, An exponential chaotic differential evolution algorithm for optimizing bridge maintenance plans, Autom. Constr. 134 (2022), https://doi.org/10.1016/j.autcon.2021.104107.

[13] Z. Sun, J.D. Xing, P.B. Tang, N.J. Cooke, R.L. Boring, Human reliability for safe and efficient civil infrastructure operation and maintenance - a review, Develop. Built Environ. 4 (2020), https://doi.org/10.1016/j.dibe.2020.100028.

[14] A.K. Ji, X.L. Xue, Q.P. Ha, X.W. Luo, M.G. Zhang, Game theory-based bilevel model for multiplayer pavement maintenance management, Autom. Constr. 129 (2021) 103763, https://doi.org/10.1016/j.autcon.2021.103763.

[15] J.S. Kong, D.M. Frangopol, Life-cycle reliability-based maintenance cost optimization of deteriorating structures with emphasis on bridges, J. Struct. Eng.-Asce 129 (2003) 818–828, https://doi.org/10.1061/(Asce)0733-9445(2003)129:6(818).

[16] M. Liu, D.M. Frangopol, Balancing connectivity of deteriorating bridge networks and long-term maintenance cost through optimization, J. Bridg. Eng. 10 (2005) 468–481, https://doi.org/10.1061/(Asce)1084-0702(2005)10:4(468).

[17] M.H. Nili, H. Taghaddos, B. Zahraie, Integrating discrete event simulation and genetic algorithm optimization for bridge maintenance planning, Autom. Constr. 122 (2021), https://doi.org/10.1016/j.autcon.2020.103513.

[18] K. Deb, An introduction to genetic algorithms, Springer 24 (1999) 293–315, https://doi.org/10.1007/bf02823145.

[19] X.M. Lei, Y. Xia, Y. Dong, L.M. Sun, Multi-level time-variant vulnerability assessment of deteriorating bridge networks with structural condition records, Eng. Struct. 266 (2022), https://doi.org/10.1016/j.engstruct.2022.114581.

[20] X. Lei, R. Feng, Y. Dong, C. Zhai, Bayesian-optimized interpretable surrogate model for seismic demand prediction of urban highway bridges, Eng. Struct. 301 (2024) 117307, https://doi.org/10.1016/j.engstruct.2023.117307.

[21] M. Liu, D.M. Frangopol, Optimizing bridge network maintenance management under uncertainty with conflicting criteria: life-cycle maintenance, failure, and user costs, J. Struct. Eng.-Asce 132 (2006) 1835–1845, https://doi.org/10.1061/(Asce)0733-9445(2006)132:11(1835).

[22] L. Miralles-Pechuán, F. Jiménez, H. Ponce, L. Martínez-Villaseñor, A deep q-learning/genetic algorithms based novel methodology for optimizing COVID-19 pandemic government actions, arXiv preprint (2020), https://doi.org/10.48550/arXiv.2005.07656.

[23] G. Ghione, V. Randazzo, A. Recchia, E. Pasero, M. Badami, Comparison of genetic and reinforcement learning algorithms for energy cogeneration optimization, in: 8th International Conference on Smart and Sustainable Technologies (SpliTech), 2023, pp. 1–7, https://doi.org/10.23919/SpliTech58164.2023.10193518.

[24] H. Ellis, M. Jiang, R.B. Corotis, Inspection, maintenance, and repair with partial observability, J. Infrastruct. Syst. 1 (1995) 92–99, https://doi.org/10.1061/(ASCE)1076-0342(1995)1:2(92).

[25] K.G. Papakonstantinou, M. Shinozuka, Planning structural inspection and maintenance policies via dynamic programming and Markov processes. Part II: POMDP implementation, Reliab. Eng. Syst. Saf. 130 (2014) 214–224, https://doi.org/10.1016/j.ress.2014.04.006.

[26] G.E. Monahan, State-of-the-art - a survey of partially observable Markov decision-processes - theory, models, and algorithms, Manag. Sci. 28 (1982) 1–16, https://doi.org/10.1287/mnsc.28.1.1.

[27] F.A. Oliehoek, C. Amato, A Concise Introduction to Decentralized POMDPs vol. 1, 2016, https://doi.org/10.1007/978-3-319-28929-8_3.

[28] M. Memarzadeh, M. Pozzi, Model-free reinforcement learning with model-based safe exploration: optimizing adaptive recovery process of infrastructure systems, Struct. Saf. 80 (2019) 46–55, https://doi.org/10.1016/j.strusafe.2019.04.003.

[29] Z.Y. Zhou, L. Lai, Y. Dong, Quantification of value of information associated with optimal observation actions within partially observable Markov decision processes, KSCE J. Civ. Eng. 26 (2022) 5173–5186, https://doi.org/10.1007/s12205-022-2121-y.

[30] R. Srinivasan, A.K. Parlikad, Value of condition monitoring in infrastructure maintenance, Comput. Ind. Eng. 66 (2013) 233–241, https://doi.org/10.1016/j.cie.2013.05.022.

[31] M. Hauskrecht, Value-function approximations for partially observable Markov decision processes, J. Artif. Intell. Res. 13 (2000) 33–94, https://doi.org/10.1613/jair.678.

[32] K.G. Papakonstantinou, C.P. Andriotis, M. Shinozuka, POMDP and MOMDP solutions for structural life-cycle cost minimization under partial and mixed observability, Struct. Infrastruct. Eng. 14 (2018) 869–882, https://doi.org/10.1080/15732479.2018.1439973.

[33] X.M. Lei, Y. Xia, L. Deng, L.M. Sun, A deep reinforcement learning framework for life-cycle maintenance planning of regional deteriorating bridges using inspection data, Struct. Multidiscip. Optim. 65 (2022) 149, https://doi.org/10.1007/s00158-022-03210-3.

[34] M. Saifullah, C.P. Andriotis, K.G. Papakonstantinou, S.M. Stoffels, Deep reinforcement learning-based life-cycle management of deteriorating transportation systems, Bridge Safet. Mainten. Manage. Life-Cycle Resilien. Sustainabil. (2022) 293–301, https://doi.org/10.1201/9781003322641-32.

[35] K. Arulkumaran, M.P. Deisenroth, M. Brundage, A.A. Bharath, Deep reinforcement learning a brief survey, IEEE Signal Process. Mag. 34 (2017) 26–38, https://doi.org/10.1109/msp.2017.2743240.
[36] V. Mnih, K. Kavukcuoglu, D. Silver, A.A. Rusu, J. Veness, M.G. Bellemare, A. Graves, M. Riedmiller, A.K. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis, Human-level control through deep reinforcement learning, Nature 518 (2015) 529–533, https://doi.org/10.1038/nature14236.
[37] A. Haydari, Y. Yılmaz, Deep reinforcement learning for intelligent transportation systems: a survey, IEEE Trans. Intell. Transp. Syst. 23 (2020) 11–32, https://doi.org/10.1109/tits.2020.3008612.
[38] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton, Y.T. Chen, T. Lillicrap, F. Hui, L. Sifre, G. van den Driessche, T. Graepel, D. Hassabis, Mastering the game of Go without human knowledge, Nature 550 (2017) 354, https://doi.org/10.1038/nature24270.
[39] S.Y. Wei, Y.Q. Bao, H. Li, Optimal policy for structure maintenance: a deep reinforcement learning framework, Struct. Saf. 83 (2020), https://doi.org/10.1016/j.strusafe.2019.101906.
[40] L.Y. Yao, Q. Dong, J.W. Jiang, F.J. Ni, Deep reinforcement learning for long-term pavement maintenance planning, Comput. Aided Civ. Inf. Eng. 35 (2020) 1230–1245, https://doi.org/10.1111/mice.12558.
[41] N.L. Zhang, W.J. Si, Deep reinforcement learning for condition-based maintenance planning of multi-component systems under dependent competing risks, Reliab. Eng. Syst. Saf. 203 (2020), https://doi.org/10.1016/j.ress.2020.107094.
[42] G. Dulac-Arnold, R. Evans, H. van Hasselt, P. Sunehag, T. Lillicrap, J. Hunt, T. Mann, T. Weber, T. Degris, B. Coppin, Deep reinforcement learning in large discrete action spaces, arXiv preprint (2015), https://doi.org/10.48550/arXiv.1512.07679.
[43] A. Tavakoli, F. Pardo, P. Kormushev, Action branching architectures for deep reinforcement learning, Proceed. AAAI Conf. Artif. Intell. 32 (2018), https://doi.org/10.1609/aaai.v32i1.11798.
[44] Z. Wang, T. Schaul, M. Hessel, H. Hasselt, M. Lanctot, N. Freitas, Dueling network architectures for deep reinforcement learning, arXiv preprint (2015) 1995–2003, https://doi.org/10.48550/arXiv.1511.06581.
[45] H. Chen, X. Dai, H. Cai, W. Zhang, X. Wang, R. Tang, Y. Zhang, Y. Yu, Large-scale interactive recommendation with tree-structured policy gradient, IEEE Trans. Knowl. Data Eng. 35 (2019) 4018–4032, https://doi.org/10.1109/TKDE.2021.3137310.
[46] Ministry of Transport of the People''s Republic of China, Standards for Technical Condition Evaluation of Highway Bridges (JTG/T H21-2011), China Communications Press Co., Ltd, Beijing, China, 2011. https://xxgk.mot.gov.cn/2020/jigou/glj/202006/t20200623_3312369.html.
[47] J. Pineau, G. Gordon, S. Thrun, Anytime point-based approximations for large POMDPs, J. Artif. Intell. Res. 27 (2006) 335–380, https://doi.org/10.1613/jair.2078.
[48] T. Tanaka, H. Sandberg, M. Skoglund, Transfer-entropy-regularized Markov decision processes, IEEE Trans. Autom. Control 67 (2022) 1944–1951, https://doi.org/10.1109/tac.2021.3069347.
[49] R.D. Smallwood, E.J. Sondik, The optimal control of partially observable Markov processes over a finite horizon, Oper. Res. 21 (1973) 1071–1088, https://doi.org/10.1287/opre.21.5.1071.
[50] G. Shani, J. Pineau, R. Kaplow, A survey of point-based POMDP solvers, Auton. Agent. Multi-Agent Syst. 27 (2013) 1–51, https://doi.org/10.1007/s10458-012-9200-2.
[51] H. Kurniawati, D. Hsu, W.S. Lee, Sarsop: efficient point-based pomdp planning by approximating optimally reachable belief spaces, *robotics*, Sci. Syst. 2008 (2008), https://doi.org/10.15607/RSS.2008.IV.009.
[52] R. Nian, J.F. Liu, B. Huang, A review on reinforcement learning: introduction and applications in industrial process control, Comput. Chem. Eng. 139 (2020) 30, https://doi.org/10.1016/j.compchemeng.2020.106886.
[53] K.G. Vamvoudakis, F.L. Lewis, Online actor-critic algorithm to solve the continuous-time infinite horizon optimal control problem, Automatica 46 (2010) 878–888, https://doi.org/10.1016/j.automatica.2010.02.018.
[54] C.P. Andriotis, K.G. Papakonstantinou, Managing engineering systems with large state and action spaces through deep reinforcement learning, Reliab. Eng. Syst. Saf. 191 (2019), https://doi.org/10.1016/j.ress.2019.04.036.
[55] O. Anschel, N. Baram, N. Shimkin, Averaged-DQN: variance reduction and stabilization for deep reinforcement learning, arXiv preprint (2016), https://doi.org/10.48550/arXiv.1611.01929.
[56] H. Van Hasselt, A. Guez, D. Silver, Aaai, deep reinforcement learning with double Q-learning, Thirtieth Aaai Conf. Artif. Intell. (2016) 2094–2100, https://doi.org/10.5555/3016100.3016191.
[57] J. Long, J. Han, Reinforcement learning with function approximation: from linear to nonlinear, arXiv preprint (2023), https://doi.org/10.48550/arXiv.2302.09703.
[58] D. Trivedi, C.D. Rahn, W.M. Kier, I.D. Walker, Soft robotics: biological inspiration, state of the art, and future research, Appl. Bionics Biomech. 5 (2008) 99–117, https://doi.org/10.1080/11762320802557865.
[59] S. Shigeno, M. Yamamoto, Organization of the nervous system in the pygmy cuttlefish, Idiosepius paradoxus ortmann (Idiosepiidae, Cephalopoda), J. Morphol. 254 (2002) 65–80, https://doi.org/10.1002/jmor.10020.
[60] L. Matignon, G.J. Laurent, N. Le Fort-Piat, Independent reinforcement learners in cooperative Markov games: a survey regarding coordination problems, Knowl. Eng. Rev. 27 (2012) 1–31, https://doi.org/10.1017/S0269888912000057.
[61] S.C. Du, Q. Deng, Q.H. Hong, J. Li, H.Y. Liu, C.H. Wang, A memristor-based circuit design and implementation for blocking on Pavlov associative memory, Neural Comput. & Applic. 34 (2022) 14745–14761, https://doi.org/10.1007/s00521-022-07162-z.
[62] D.P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint (2014), https://doi.org/10.48550/arXiv.1412.6980.
[63] T. Schaul, J. Quan, I. Antonoglou, D. Silver, Prioritized experience replay, arXiv preprint (2015), https://doi.org/10.48550/arXiv.1511.05952.
[64] F.Q. Zhao, F.Q. Liu, Z.W. Liu, H. Hao, The correlated impacts of fuel consumption improvements and vehicle electrification on vehicle greenhouse gas emissions in China, J. Clean. Prod. 207 (2019) 702–716, https://doi.org/10.1016/j.jclepro.2018.10.046.
[65] A.K. Agrawal, A. Kawaguchi, Z. Chen, Deterioration rates of typical bridge elements in New York, J. Bridg. Eng. 15 (2010) 419–429, https://doi.org/10.1061/(ASCE)BE.1943-5592.0000123.
[66] T.L. Cavalline, M.J. Whelan, B.Q. Tempest, R. Goyal, J.D. Ramsey, Determination of Bridge Deterioration Models and Bridge User Costs for the NCDOT Bridge MANAGEMENT system, No. FHWA/NC/2014-07. https://trid.trb.org/view/1405296, 2015.
[67] N.K.W. Wellalage, T.L. Zhang, R. Dwight, Calibrating Markov chain-based deterioration models for predicting future conditions of railway bridge elements, J. Bridg. Eng. 20 (2015), https://doi.org/10.1061/(asce)be.1943-5592.0000640.
[68] G. Qiao, X. Gao, K. Ren, J. Tao, Research on the application of technology of replacing bridge bearings without interrupting traffic, J. Phys. Conf. Ser. 2021 (1865) 032041, https://doi.org/10.1088/1742-6596/1865/3/032041.
[69] Ministry of Housing and Urban-Rural Development of the People's Republic of China, Code of valuation with bill quantity of construction works (GB50500-2013), China Planning Press Co., Ltd, Beijing, China, 2013. https://sj.ujs.edu.cn/info/1123/2069.htm.
[70] X.M. Lei, Y. Dong, Deep reinforcement learning for optimal life-cycle management of deteriorating regional bridges using double-deep Q-networks, Smart Struct. Syst. 30 (2022) 571–582, https://doi.org/10.12989/sss.2022.30.6.571.
[71] Ministry of Transport of the People''s Republic of China, Design Specification for Highway Alignment (JTG D20-2017), China Communications Press Co., Ltd, Beijing, China, 2017. https://xxgk.mot.gov.cn/2020/jigou/glj/202006/t20200623_3312660.html.
[72] M. Barbehenn, A note on the complexity of Dijkstra's algorithm for graphs with weighted vertices, IEEE Trans. Comput. 47 (1998) 263, https://doi.org/10.1109/12.663776.
[73] L. Elefteriadou, The highway capacity manual 6th edition: a guide for multimodal mobility analysis, Ite. J. 86 (2016) 14–18. https://trid.trb.org/view/1403977.
[74] H. Bourenane, M.S. Guettouche, Y. Bouhadad, M. Braham, Landslide hazard mapping in the Constantine city, Northeast Algeria using frequency ratio, weighting factor, logistic regression, weights of evidence, and analytical hierarchy process methods, Arab. J. Geosci. 9 (2016) 1–24, https://doi.org/10.1007/s12517-015-2222-8.
[75] X. Huang, S.J. Xiao, Acm, self-augmenting strategy for reinforcement learning, in: Proceedings of the 2017 International Conference on Computer Science and Artificial Intelligence (Csai 2017), 2017, pp. 1–4, https://doi.org/10.1145/3168390.3168392.