Nuclear Engineering and Technology 56 (2024) 1959–1974

Contents lists available at ScienceDirect

Nuclear Engineering and Technology

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

Review Article

Possibilities of reinforcement learning for nuclear power plants: Evidence on current applications and beyond☆

Aicheng Gong [a,b,1], Yangkun Chen [b,1], Junjie Zhang [b,1], Xiu Li [b,*]

[a] State Key Laboratory of Nuclear Power Safety Monitoring Technology and Equipment, China Nuclear Power Engineering Company Ltd., Shenzhen 518172, China
[b] Shenzhen International Graduate School, Tsinghua University, Shenzhen 518055, China

ARTICLE INFO

Keywords:
Nuclear Power Plant
Complex system
Reinforcement learning
Control
Artificial intelligence

ABSTRACT

Nuclear energy plays a crucial role in energy supply in the 21st century, and more and more Nuclear Power Plants (NPPs) will be in operation to contribute to the development of human society. However, as a typical complex system engineering, the operation and development of NPPs require efficient and stable control methods to ensure the safety and efficiency of nuclear power generation. Reinforcement learning (RL) aims at learning optimal control policies via maximizing discounted long-term rewards. The reward-oriented learning paradigm has witnessed remarkable success in many complex systems, such as wind power systems, electric power systems, coal fire power plants, robotics, etc. In this work, we try to present a systematic review of the applications of RL on these complex systems, from which we believe NPPs can borrow experience and insights. We then conduct a block-by-block investigation on the application scenarios of specific tasks in NPPs and carried out algorithmic research for different situations such as power startup, collaborative control, and emergency handling. Moreover, we discuss the possibilities of further application of RL methods on NPPs and detail the challenges when applying RL methods on NPPs. We hope this work can boost the realization of intelligent NPPs, and contribute to more and more research on how to better integrate RL algorithms into NPPs.

1. Introduction

As the world's population grows, the demand for energy for human production and survival increases considerably [1–3]. More and more greenhouse gases are produced into the atmosphere to meet the energy demands of human society, resulting in climate change and the greenhouse effect. To mitigate the effects of climate change, the world must rapidly reduce its reliance on fossil fuels and its emissions of greenhouse gases. Nuclear energy is low-carbon and can be installed on a massive scale within the required timeframe, providing the world with clean, reliable, and inexpensive electricity. Nuclear Power Plants (NPPs) [4–6] emit no greenhouse gases during operation, and over the course of its life cycle, nuclear fuel produces roughly the same amount of carbon dioxide-equivalent emissions per unit of electricity as wind and one-third of the emissions per unit of electricity when compared to solar.

"The Report On The Development of China's Nuclear Energy 2021" [7] shows that during the "13th Five-Year Plan" period, China's nuclear power units will maintain safe and stable operation. Currently, 20 new commercial nuclear power units have been put into operation. The newly installed capacity has reached 23.447 million kilowatts, and the total number of commercial nuclear power units has reached 48 units. The installed capacity is 49.88 million kilowatts, ranking third in the world in installed capacity and the second in power generation in 2020. 11 new nuclear power units with an installed capacity of 12.604 million kilowatts have been built, ranking first in the world in terms of the number of units under construction and installed capacity for many years.

In 2020, China's nuclear power generation reached 366.243 billion kWh, with an increment of 5.02% year-on-year, accounting for about 4.94% of the country's cumulative power generation. Compared with coal-fired power generation, the annual nuclear power generation is equivalent to reducing the burning of standard coal by 104.7419 million tons, reducing the emission of 274.4238 million tons of carbon dioxide (CO_2), 890,300 tons of sulfur dioxide, and 775,100 tons of nitrogen oxides, equivalent to afforestation of 771,400 hectares. Over

☆ This work is supported by the SIT 2030-Key Project under Grant 2021ZD0201404.
* Corresponding author.
E-mail addresses: gongaicheng@cgnpc.com.cn (A. Gong), chen-yk21@mails.tsinghua.edu.cn (Y. Chen), zhangjj21@mails.tsinghua.edu.cn (J. Zhang), li.xiu@sz.tsinghua.edu.cn (X. Li).
[1] These authors contribute equally to this work.

https://doi.org/10.1016/j.net.2024.01.003
Received 8 August 2023; Received in revised form 7 December 2023; Accepted 1 January 2024
Available online 4 January 2024

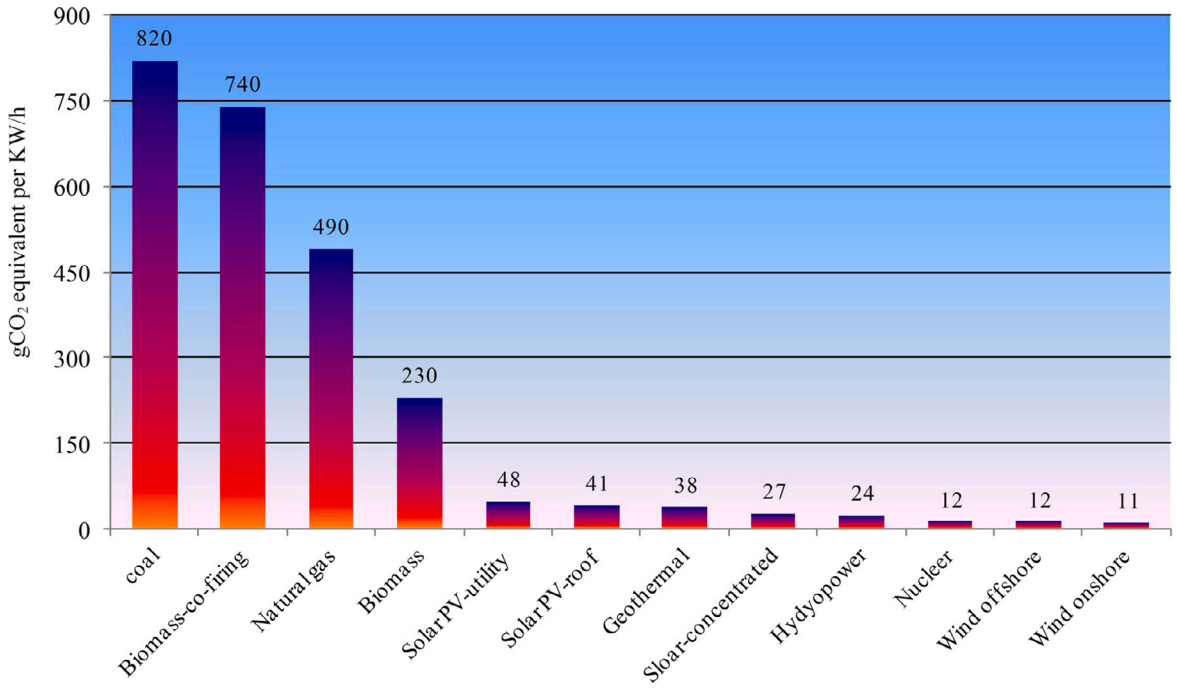

Fig. 1. Average life-cycle carbon dioxide-equivalent emissions for different electricity generators [8].

the past decade, nuclear power generation has continued to grow, making important contributions to ensuring power supply security and energy conservation, and emission reduction [7].

During the "14th Five-Year Plan" and in the medium and long term, China's nuclear power will shift to a new stage of positive and orderly development on the premise of ensuring safety. Under the background of carbon peaking and carbon neutrality, the clean and low-carbon transformation process of China's energy and power system will be further accelerated. As a clean energy with near zero emissions shown in Fig. 1, nuclear energy has a broad possibility of being further developed and publicized. The technology on nuclear energy is also expected to maintain rapid development. In the current situation, China's independent third-generation nuclear power will achieve large-scale batch development according to the approval rhythm of 6 to 8 units per year. It is estimated that by 2025, the installed capacity of nuclear power in China will be about 70 million kilowatts, along with about 50 million kilowatts being under construction.

In the last 50 years, from 70 gigatonnes (Gt) to 78 Gt CO_2 emission has been effectively avoided with the service of NPPs globally. At the end of 2020, more than 400 nuclear power reactors are working safely in 32 countries, with a total power capacity of 392.6 GW. Even during the coronavirus disease (COVID) pandemic, none of the operating NPPs in these 32 countries with operating NPPs reported that the pandemic had induced an operational event that may impact the safety and reliability of the NPP operation. In 2020, nuclear power supplied 2553.2 terawatt-hours of GHG emission-free electricity accounting for about 10% of total global electricity generation and nearly a third of the world's low carbon electricity production.

As the key to the success of the decarbonization of the electricity sector, nuclear power provides reliable low-carbon power to the grid around the clock. With the global increase in electricity demand to satisfy the needs of the world's population, nuclear power will be necessary.

The utilization and promotion of NPPs require a high technical threshold, which brings great challenges to the popularization of NPPs and electricity generation with nuclear energy. As a typical man-machine-network integration system [9–11], great complexity is shown in many aspects of the nuclear industry.

In NPPs, there are many operations that require a lot of decision-making. At the current stage, traditional manual operations or more classical control methods are widely used, such as proportional-integral-differential (PID) controllers [12–18], programmable logic controllers (PLCs) [19–22], and field-programmable gate arrays (FPGAs) [23–25]. Traditional PID control is often designed for a single subsystem with a single control variable [26]. In the more complex application scenario of NPPs, indicators of the PID controller, such as overshoot and response speed, are often unsatisfactory. A large number of power plant parameters need to be tested and monitored, which poses many challenges to operators and traditional controllers. From this perspective, it is imperative to introduce intelligent processing and control methods into NPPs, which is also emphasized in a recent paper [27].

In recent years, with the enhancement of computer computing capability, the generation and collection of a large amount of data, and the proposal of new algorithms, deep learning has made great progress and has exerted its own advantages in many fields [28–34]. Among them, reinforcement learning (RL) [35–37] is a typical and important machine learning method. Deep reinforcement learning (DRL), which combines RL with deep learning (e.g., deep neural networks), shines in the fields of robots, wind turbines, and fire control. This is because RL requires constant interaction with the environment in its mechanism design. The RL agent learns the best policy under specific observations during the process of continuous trial and error, aiming at maximizing the cumulative returns. RL can implicitly model many complex problems. As a class of systems that are widely studied, complex systems have strong coupling relationships among their parts, and the controlled objects often contain multiple variables and have the characteristics of nonlinearity. At this time, RL can achieve better performance in the face of random sequential decision-making tasks, so it has been widely used in many complex system tasks, such as wind, fire, coal, power grids, and robots.

We set our focus firstly on complex systems because there is currently little research on integrating RL algorithms into NPPs. Meanwhile, NPPs are typically complex systems in nature, and there are many studies on combining RL methods with complex systems. We, therefore, believe that some strongly related reviews on complex systems can bring some new insights into the future application of RL algorithms in NPPs.

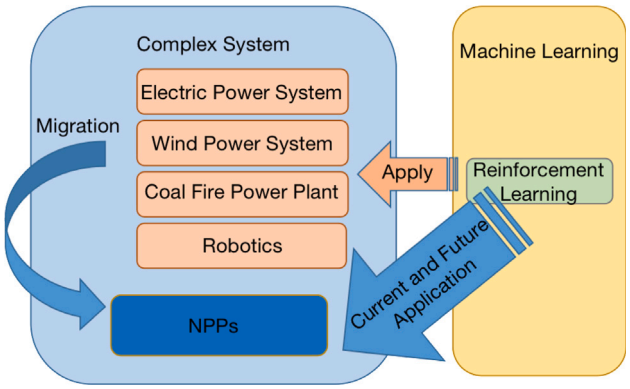

Fig. 2. Main structure of this paper.

1.1. Our contribution and organization

This paper mainly investigates and summarizes the main application of reinforcement learning, a widely used algorithm model in machine learning, in NPPs. Considering the extensive and exciting applications of existing RL work in various complex system tasks, we first investigate the use of RL algorithms in power grids, heat energy, wind energy, and robotics. We then investigate the relationship between complex systems and NPPs, where we believe that the successful and wide application of RL on these complex systems can bring some insights to NPPs. We further explore how we will apply existing RL algorithms to various parts of NPPs under the existing models to promote the construction of future smart factories. Our main contributions are:

- We introduce mainstream RL algorithms and frameworks, and point out the advantages and limitations of RL algorithms compared to traditional control algorithms.
- We also research the application of RL in various complex systems and combine it with the existing work to put forward a further prospect for the migration of RL to NPPs in the future.
- We investigate the application of RL in NPPs today, classify them according to various functions and modules of NPPs, and explore the feasibility of RL algorithms in these tasks.

This article will be expanded in the following aspects (see Fig. 2): Section 2 will investigate and introduce the mainstream RL methods. We classify them into value-based methods and policy-based methods and point out the advantages and disadvantages of different algorithms. Section 3 investigates the related applications of RL algorithms in complex systems, including power grids, coal, wind power, and robotics. Through the study of these applications, we can further pave the way and introduce the application of RL algorithms in NPPs. Section 4 investigates various applications of RL methods in NPPs, where we classify them according to different processes or tasks of NPPs, such as power startup, thermal control, cooperative operation, emergency handling processes, etc. Section 5 discusses and studies the future development of NPPs and how RL will play a further role in the NPPs. Section 6 concludes with a well-rounded description to better conduct the whole part of the article.

As there are many abbreviations in this article, we refer readers to a checklist that explains the meanings of adopted abbreviations in Table 3.

2. Background

2.1. Complex system

Industrial complex systems generally refer to large-scale systems represented by water energy, wind energy, power grids or robots [38,39]. The complexity is determined by the number of components, the complexity of interfaces, and the degree of nesting in structure components, which are often more difficult to analyze and control than simple systems [40]. It has the following characteristics:

- **Nonlinearity**: The degree of linearity is often regarded as an important condition for measuring the degree of complexity [41]. Generally speaking, we cannot use the superposition method to control the system that does not have linear properties, and the nonlinearity in the control equation manifests that a slight deviation in the initial condition value can lead to a completely different macroscopic result [42].
- **Feedback ability**: This kind of feedback is not only reflected in the design of complex systems with feedback modules for external input, but more importantly, the mutual feedback function between internal individuals [43]. Taking the complex system of the robot as an example, each part of the robot will adjust its own motion according to other parts, and its adjacent parts can partially reflect its early behavior of it. By means of feedback, all low-level parts can maintain a higher level of order and achieve more complex functions.
- **Order**: Complex systems can spontaneously maintain high-level order, which means that a large number of uncoordinated interactions between systems can demonstrate a spontaneous and rational order in general [44].
- **Robustness**: Complex systems often have strong robustness, which is reflected in their distributed functions [45]. Most complex systems have a central control system. The central control system often has good robust performance, and when a general non-central system fails, the impact on the entire system will not be too large.
- **Hierarchical structure**. Complex systems are generally considered to have many levels of organization that can be used to form the structure of the system [46], and the order of interactions between lower-level structures is robust.

2.2. Reinforcement learning

Reinforcement learning (RL) is a typical method in artificial intelligence (AI), which requires the entity (referred to as an agent in RL) to interact with the environment to receive the rewards, and therefore renew its behavior or policy to better maximize the expected return [35]. Recently RL has achieved remarkable performances in different areas such as games or robotics [47–56]. RL problems can be typically described by a Markov Decision Process (MDP). The basic components of the MDP include state set $\mathcal{S}$ representing environmental information that is vital for decision-making, action set $\mathcal{A}$ representing actions that the agent can take, environmental dynamics $p(s'|s,a)$ representing the probability of transitioning from state s to next state s', discount factor $\gamma \in [0,1)$, and scalar reward signal r for an agent. Based on the current state s, the agent selects an action a from all the potential behavior sets of $\mathcal{A}$ to act in the environment. After the environment accepts the action, it changes and generates a reward signal r to feedback to the agent. The agent then continuously selects the next action according to the signal and the current state of the environment [35].

Deep learning is a concept that has produced extraordinary results in a variety of fields [29,32,57–63]. It relies on the neural network to approximately fit a nonlinear function. Deep reinforcement learning (DRL), when paired with the aforementioned technologies, is a potent, widely applicable technology [28,37,64–68].

DRL can often be divided into two distinct categories. The first way is value-based, which requires a value evaluation for each action-state pair. The second approach is the policy-based method, the primary premise of which is to determine the optimal policy or behavior given the existing condition.

2.2.1. Value-based method for the DRL

To better demonstrate the idea of value-based DRL, we first introduce the basic idea of Q-learning, which lays the foundations for the recent DRL algorithms.

The Q-learning algorithm is an RL algorithm based on the value function [69]. The Q-learning algorithm attempts to build and maintain a Q-table to measure the expected long-term rewards of each pair of state and action. In this algorithm, we need to update the Q-values in the Q-table according to Eq. (1) below. Where α is the learning rate and γ is the decay factor, $\gamma \in [0, 1)$. Controlling the size of the two factors can adjust the learning mode of the agent. When γ is set to be larger, the agent will review the previous learning situation more. And when γ is set smaller, the influence of the current state is more obvious.

$$Q\left(s_t, a_t\right) \leftarrow Q\left(s_t, a_t\right) + \alpha\left[r_{t+1} + \gamma \max_{a \in \mathcal{A}} Q\left(s_{t+1}, a\right) - Q\left(s_t, a_t\right)\right]. \tag{1}$$

DQN: When the situation becomes complex, which means there are bounds of possible states or even the state space becomes continuous, maintaining a complete, accurate Q-table is almost impossible. With the development of deep learning algorithms, researchers have noticed that representing the Q function with the neural network can effectively solve more complex environments and tasks (not just tabular MDP). That is, we use the powerful fitting ability of the neural network to enhance the representation capability of the agent and incur a better function approximation. Based on this idea, scholars propose the Deep Q-Network (DQN) algorithm [47,70].

Eq. (2) is called the Bellman optimal equation:

$$q_*(s, a) = \mathcal{R}_s^a + \gamma \sum_{s' \in \mathcal{S}} \mathcal{P}_{ss'}^a \max_{a'} q_*\left(s', a'\right). \tag{2}$$

Here we use this equation to bridge our predicted value with the target value, and the optimization function can be withdrawn as the difference between them. During the update process of the Q-value, if the same network is used to predict the Q-value of the current state and the next state, the network output may become unstable. Previous studies have shown that this phenomenon is more pronounced when nonlinear activation functions are used [71,72]. Considering this, DQN proposes a dual network structure, that is to decouple the calculation of the predicted Q-value and the target Q-value: one of the networks is used to output the predicted Q-value, and the other network (which is a lagging network that synchronizes parameters from the current network every fixed gradient steps) is based on the reward to get the target Q-value. Moreover, in order to solve the problem of the correlation of the input data and the uncertainty of the output distribution, the DQN algorithm implements the experience replay [73]. The main idea is that the data obtained by the interaction between the agent and the environment in each state is recorded as a sample and stored in the data pool. When training a neural network, a small part of the data is randomly selected from the data pool and sent to the network as a mini-batch for learning.

Accordingly, the loss function of the network is changed to Eq. (3) below. Where θ is the parameter of the current network that predicts the Q-value and θ^- is the parameter of the lagging network that outputs the target Q-value.

$$\mathcal{L}(\theta) = \left(r_{t+1} + \gamma \max_{a'} Q_{\theta^-}\left(s_{t+1}, a'\right) - Q_\theta\left(s_t, a_t\right)\right)^2. \tag{3}$$

The whole training procedure is shown in Fig. 3.

Double DQN: Along with the DQN algorithm's successful combination of RL algorithms and deep neural networks, researchers have also discovered corresponding problems. When DQN updates the Q-value using the Bellman optimal equation, the action corresponding to the maximum Q-value in the next state is selected each time, and the selection and evaluation of values upon actions are based on the parameters of the target value network. It will lead to the problem of overestimating the Q-value. To tackle this problem, reviewing the idea of DQN in dealing with unstable network output and based on the idea of double Q-learning [74], the researchers propose the double network structure for the decoupling of Q-value estimation and action selection, and they propose the Double DQN model [75]. At this time, the loss function of the network is updated to Eq. (4) below.

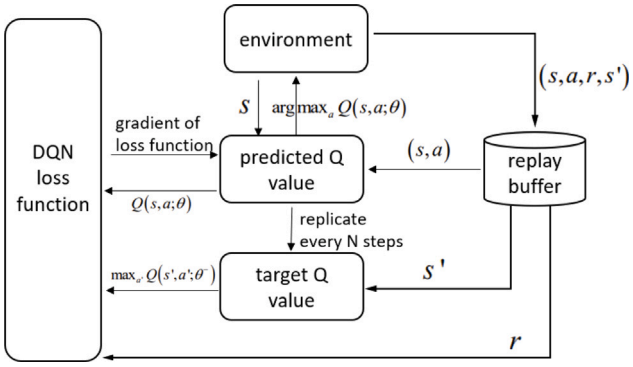

Fig. 3. Graph illustration of the training procedure of DQN.

$$\mathcal{L}(\theta) = \left(r_{t+1} + \gamma Q_{\theta^-}\left(s_{t+1}, \operatorname{argmax}_{a'} Q_\theta\left(s_{t+1}, a'\right)\right) - Q_\theta\left(s_t, a_t\right)\right)^2. \tag{4}$$

2.2.2. Policy-based method for the DRL

Compared to RL algorithms based on the value function, policy gradient algorithms, in which the neural network directly outputs the corresponding action, are more prevalent in the contemporary industry. In discrete action spaces, RL approaches based on value functions have been widely applied, however, in continuous action spaces, we cannot derive state–action value functions for each state. In this situation, policy-based solutions are more advantageous.

VPG: The core idea of the Vanilla Policy Gradient (VPG) algorithm is to parameterize the behavior policy, calculate the policy gradient about the action, and continuously adjust the action along the gradient direction. VPG ensures that the updated policy could be better than the old policy, thus continuing to converge to the optimal policy. Categories in the policy gradient can be divided into stochastic policy $a \sim \pi_\theta(a|s) = P(a|s;\theta)$ and deterministic policy $a = \mu_\theta(s)$. The former returns the probability of the action under a certain state, while the latter corresponds to the deterministic action. The core concept of policy gradient is to increase the probability of actions that lead to higher rewards while decreasing the probability of actions that lead to lower rewards until an optimal policy is reached. Sutton et al. [76] summarized previous work on policy gradient in 2000, which laid the foundation for modern policy gradient methods; Schulman et al. [77] propose the concept of generalized advantage function estimation and gave a relatively well-established problem description for policy gradients. Mathematically formulating the problem is shown below.

For a given policy network, our goal is to maximize the expected return by adjusting the parameter θ:

$$J\left(\pi_\theta\right) = \mathbb{E}[r(\tau)], \tag{5}$$

where τ is the trajectory, i.e., $\tau = \langle s_0, a_0, s_1, a_1, \ldots, s_N, a_N \rangle$.

For the expected return, by introducing the relevant characteristics of the log derivation calculation, the results we obtain after derivation

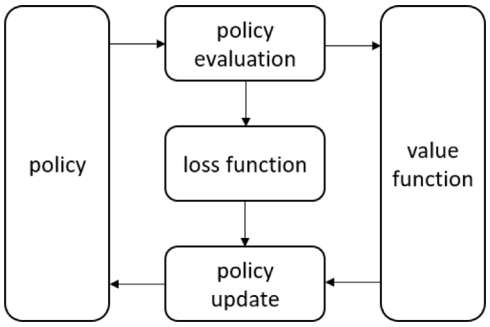

Fig. 4. The structure of Actor-Critic methods.

are as follows:

$$\begin{aligned}\nabla_\theta J\left(\pi_\theta\right) &= \nabla_\theta \mathbb{E}_{\tau\sim\pi_\theta}[r(\tau)] \\ &= \nabla_\theta \int_\tau P(\tau \mid \theta) r(\tau) \\ &= \int_\tau \nabla_\theta P(\tau \mid \theta) r(\tau) \\ &= \int_\tau P(\tau \mid \theta) \nabla_\theta \log P(\tau \mid \theta) r(\tau) \\ &= \mathbb{E}_{\tau\sim\pi_\theta}\left[\nabla_\theta \log P(\tau \mid \theta) r(\tau)\right].\end{aligned} \tag{6}$$

By using the sampled trajectory to represent the expectation, we can get the final optimization function:

$$\hat{g} = \frac{1}{|D|} \sum_{\tau\in D} \sum_{t=0}^{T} \nabla_\theta \log \pi_\theta\left(a_t \mid s_t\right) r(\tau), \tag{7}$$

where D In contrast to supervised learning, the loss function is defined and its value is supposed to be minimized, and the policy gradient scenario seeks to maximize the expected reward function. Through detailed inspection of the formula, it is possible to conclude that $R(tau)$ represents the current state–action pair's weight. When the reward is low, it is reflected in the expected reward function, making the action less likely to be chosen; otherwise, the action is more likely to be chosen. The future series of gradient-based policy techniques may incorporate more enhancements to this formula. Then, an additional difficulty is posed, which is to assume that all rewards under the current job are bigger than zero so that the chance of occurrence of the inferior trajectory will always rise. The strategy for overcoming this problem is to substitute the reward value with an advantage function. The advantage function is defined as the current action value function minus the current state value function, therefore it can indicate the degree to which the present behavior is superior to the average. How to define the current state value function remains to be determined. To estimate the state value, an extra network is provided as a solution. This is also the basic concept of the Actor-Critic method, which employs two networks to return the output action and the current state value, respectively. The entire procedure is depicted in Fig. 4.

DDPG: Deep Deterministic Policy Gradient (DDPG) [49] can generate a deterministic policy. It combines the ideas of DQN and VPG, and a total of 4 networks are used for model learning. The calculation formula of DPG's deterministic policy gradient based on Q-value is as follows:

$$\nabla_\theta J\left(\pi_\theta\right) = E_{s\sim\rho^\pi}\left[\nabla_\theta \pi_\theta(s) \nabla_a Q_\pi(s,a)\big|_{a=\pi_\theta(s)}\right] \tag{8}$$

The transition from DPG to DDPG can be analogous to the transition from DQN to DDQN, that is, a dual network is added, which has both the current network and the target network. Since DDPG adopts the Actor-Critic structure, we will finally get four networks remarked as actor current network, actor target network, critic current network, and the critic target network. The actor's current network is used to iteratively update the policy network parameters θ, select the current action a according to the current state s, and interact with the environment to generate s' and r; The Actor target network is responsible for selecting the optimal next action a' according to the next state s' sampled in the replay pool based on experience, and the network parameter θ^- is periodically copied from θ. The critic current network is used for the iterative update of the value network parameter w, and is responsible for calculating the current Q-value; The critical target network is responsible for computing the target Q-value, while the network parameter w^- is periodically copied from w. On the basis of the DDPG algorithm, Popov et al. [78] further expand the algorithm from two aspects and propose an asynchronous DPG method. Specifically, the authors increased the number of iterative updates learned after the agent interacts with the environment, in a way that reduces the number of interactions with the environment required to learn a successful policy.

TRPO&PPO: The previously introduced VPG algorithm has a significant flaw, namely, if the network parameters are updated, the goal function will also change, as the objective function is an expression connected to *theta*, necessitating the collection of new samples. In addition, the VPG method has a fatal fault in that the step size of each update is fixed, making it easy for a policy to degrade after being updated. In order to solve this problem, Schulman et al. [79] proposed the Trust Region Policy Optimization (TRPO) algorithm, which uses confidence intervals for correlation constraints and theoretically proved that the new policy must be better than the old one. The author proposed,

$$\eta(\tilde{\pi}) = \eta(\pi) + \mathbb{E}_{s_0,a_0,\ldots\sim\tilde{\pi}}\left[\sum_{t=0}^{\infty} \gamma^t A_\pi\left(s_t, a_t\right)\right] \tag{9}$$

Thus, the new policy might always be improved so long as the second term is guaranteed to be bigger than zero. The previous policy state distribution can be utilized to estimate the new policy state distribution via importance sampling. In addition, KL divergence is utilized to limit the difference between the old and new policies so that it does not become excessively huge. The new formula for optimization is:

$$\begin{aligned}&\max_\theta \mathbb{E}_{s\sim\rho_{\theta_{old}}, a\sim\pi_{\theta_{old}}}\left[\frac{\tilde{\pi}_\theta\left(a \mid s_n\right)}{\pi_{\theta_{old}}\left(a \mid s_n\right)} A_{\theta_{old}}\left(s_n, a\right)\right], \\ &\text{s.t.} \quad D_{KL}^{\max}\left(\theta_{old} \parallel \theta\right) \le \delta.\end{aligned} \tag{10}$$

The Proximal Policy Optimization (PPO) algorithm is obtained by refining the TRPO method further [80]. The essential idea of PPO is to use the clip function to restrict the coefficients between update policies so that, under varying conditions, the estimation of the coefficients in front of the advantage function is restricted to a set range to prevent over-optimization.

$$\begin{aligned}&if \quad \hat{A}_t > 0, \\ &L^{clip}(\theta) = \begin{cases}(1+\epsilon)\hat{A}_t & r_t(\theta) > 1+\epsilon, \\ r_t(\theta)\hat{A}_t & \text{else.}\end{cases} \\ &if \quad \hat{A}_t < 0, \\ &L^{clip}(\theta) = \begin{cases}(1-\epsilon)\hat{A}_t & r_t(\theta) < 1-\epsilon, \\ r_t(\theta)\hat{A}_t & \text{else.}\end{cases}\end{aligned} \tag{11}$$

A3C: A3C (Asynchronous advantage actor-critic) is a novel algorithm developed by Google DeepMind to tackle the Actor-Critic non-convergence problem [81]. An essential difficulty in RL is how to decorrelate the reliance between data. A3C's proposed approach is to employ an asynchronous technique. In general, A3C generates many parallel environments and permits various agents with secondary structures to change parameters in the primary structure simultaneously in these parallel settings. Notably, parallel agents do not interfere with one another during training, and the discontinuity in the update supplied by the secondary structure will disrupt the parameter update of the primary structure, thereby reducing the correlation of the update

and improving convergence. The essence of the A3C method during the training process is to place the Actor-Critic in numerous threads for synchronous training, while a central processing unit summarizes the training results and provides the modified gameplay back to these multiple threads. These workers may collect data with minimal sample correlation and learn, which is also the basic concept underlying A3C's series of algorithms.

SAC: The Soft Actor-Critic (SAC) algorithm [82] aims to learn a policy that not only maximizes the cumulative reward but also the entropy of the policy. The incorporation of entropy maximization promotes exploration by the agent, enhancing the policy's robustness and effectiveness in diverse scenarios. The objective function of SAC, reflecting this balance, is given by:

$$J(\pi) = \mathbb{E}_{(s_t,a_t)\sim\rho_\pi}\left[\sum_t \left(r(s_t,a_t) + \alpha\mathcal{H}(\pi(\cdot|s_t))\right)\right] \tag{12}$$

Here, $J(\pi)$ denotes the objective function in terms of policy π, $r(s_t,a_t)$ is the reward function, $\mathcal{H}(\pi(\cdot|s_t))$ represents the entropy of the policy at state s_t, and α is a temperature parameter that balances the importance of the entropy term relative to the reward. The policy π is derived to optimize this objective, and ρ_π indicates the state–action distribution under policy π.

The SAC algorithm utilizes two Q-functions to stabilize the training process and minimize the overestimation bias, a common issue in value-based methods. It employs a reparameterization trick for efficient policy gradient updates, contributing to its stability and convergence properties. Overall, SAC has been demonstrated to perform exceptionally well in various continuous control tasks, outperforming many classical RL algorithms. Its unique combination of off-policy learning and entropy-based exploration strategy makes it highly suitable for tasks with complex, high-dimensional action spaces.

MPO: The Maximum a Posteriori Policy Optimization (MPO) [83] algorithm is a sophisticated RL approach, predominantly employed for solving complex control problems in continuous action spaces. It is a policy-based method that focuses on enhancing policy stability and reliability while maintaining performance efficiency. MPO combines classic policy gradient techniques with contemporary policy optimization strategies, employing Maximum a Posteriori (MAP) estimation to refine the policy.

The core idea of MPO lies in its dual optimization process during policy iteration. Initially, the algorithm estimates the posterior probability distribution of actions using a collected set of data, aiding in a better understanding of the current policy's behavior across various states. Subsequently, it applies MAP estimation for policy updating, aiming to achieve higher expected rewards. One of the primary advantages of the MPO algorithm is its effective balance between exploration and exploitation, ensuring stability in policy updates. This characteristic renders it exceptionally suitable for problems involving high-dimensional state and action spaces, particularly in complex robotic control and simulation environments. Through meticulous policy updates and efficient utilization of posterior probabilities, MPO achieves robust performance improvements while maintaining high sample efficiency.

2.2.3. Advanced topics in DRL

In this subsection, we introduce some advanced topics in DRL that have some potential applications in NPPs.

Offline RL. Different from online RL, offline RL aims at learning optimal policies from some static offline datasets, which were previously gathered by some unknown behavior policy. Offline RL eliminates the need of getting access to the environment to train RL algorithms, which is extremely appealing to NPPs due to safety considerations. A central challenge in offline RL is extrapolation error [84], where the value estimate can be inaccurate upon unseen state–action pairs. To address this, existing methods can be categorized into value pessimism methods [84–86], policy constraint methods [87–89], etc.

Model-based RL. Model-based RL [90] optimizes its policy with the aid of the learned dynamics model, which models the physical characteristics in the system. Its application in NPPs is also appealing since we can learn the key physical dynamics via neural networks, and use the dynamics models to generate synthetic transitions. Then those imagined transitions (i.e., imagined *fake* samples) can be used for training policies, enabling the policy to learn faster and better. The core benefit of using model-based RL is its superior sample efficiency. There are many interesting researches in model-based RL, including utilizing it in the offline setting [91–94], how to learn a better model [95,96], etc.

3. Reinforcement learning on complex systems

There are many basic applications of RL on complex systems. In the power supply held on a complex system, for example, Zhou et al. [97] propose an AI agent that was based on DRL for handling various operating scenarios for the economic dispatch of a combined heat and power system. In energy management, Samadi et al. [98] propose the use of decentralized multiagent systems (MASs) for integrated grid-connected microgrids. MASs with DRL has shown not only flexible management while considering customer consumption but also reduced operating costs. Kazmi et al. [99] optimize the energy efficiency of hot water production by using a DRL controller, which could reduce the energy consumption by almost 20% for a set of 32 Dutch houses. This method also significantly reduces the energy cost of an HVAC (heating, ventilation, and air conditioning) system by using DRL instead of rule-based and model-based strategies [100]. In another study, DRL is adopted in urban rail transit to effectively improve energy management compared to the genetic algorithms. In the following part, we would like to introduce some general applications on several typical complex systems of RL. Although most experiments are conducted on simulation platforms, the application of RL to complex systems can still provide valuable insights for implementing RL in nuclear power plants.

To facilitate the readers to better capture the complex systems and algorithms used in these systems, we present in Table 1 a summary of papers on the typical complex systems, the operating areas, the utilized RL algorithms, and the published years.

3.1. Wind power system

As one of the applications of RL algorithms in the field of the wind power complex system, in the research conducted by Oh et al. [101], RL algorithm is implemented in ESS (Energy Storage Systems) operation to manage the WPG (Wind Power Generation) forecast uncertainty, which is a critical challenge for wind power generation. To handle these questions, the ESS operation problem is presented as the MDP model with the ES (Energy subsystem) and PS (Power subsystem) constraints. The SARSA-based algorithms [35] and Q-learning-based algorithms are implemented to find the optimal policy of the MAE (Mean Absolute Error) minimization problem based on MDP. In this task, the SARSA-based method gives a more robust performance compared to the method based on Q-learning. Results of simulations conducted based on practical WPG generation data and forecasting indicate that the proposed RL-based ESS operation strategy can manage the WPG forecast uncertainty more effectively than conventional Q-learning-based methods.

Wei et al. [102] establish a RL-based intelligent maximum power point tracking (MPPT) algorithm for variable-speed wind energy conversion systems (WECSs). By updating the action values recorded in a Q-table, a model-free Q-learning algorithm is intended to determine the optimal policy for the controller of the WECS. After a period of online learning, the maximum power points (MPPs) are determined in order to produce an optimal speed-power curve for rapid MPPT control

Table 1
An outline of reinforcement learning on a complex system.

Complex systems	Operating areas	RL algorithm	Year
Wind power system			
Oh et al. [101]	ESS(Energy Storage Systems)	SARSA-based,Q-learning-based	2020
Wei et al. [102]	WECSs(Wind Energy Conversion Systems)	Q-learning-based	2015
Zhang et al. [103]	Wind power prediction	DDPG	2021
Electric power systems			
Xu et al. [104]	Reactive power control	Distributed Q-learning	2012
John et al. [105]	Reactive Power Control	Q-learning	2004
Yu et al. [106],	AGC(Automatic Generation Control)	Q-learning	2011
Daneshfar et al. [107],	AGC(Automatic Generation Control)	Q-learning	2010
Ahamed et al. [108],	AGC(Automatic Generation Control)	Q-learning	2002
Yu et al. [109],	AGC(Automatic Generation Control)	Q-learning	2012
Ye et al. [110]	Market decision	DDPG	2019
Zarrabian et al. [111]	Generators power control	Q-learning	2016
Liu et al. [112]	Cyber-physical security assessment	Q-learning	2019
Coal-fired power plant			
Cheng et al. [113]	Coal-fired boilers combustion optimization system	DQN	2018
Fu et al. [114]	Denitrification system	A3C	2020
Zhan et al. [115]	Combustion control	MORE	2021
Stephan et al. [116]	Industrial hard-coal combustion process control	Multiagent system	2001
Robotics			
Andrychowicz et al. [117]	Migration from simulation to real objects	Distributed RL	2020
Akkaya et al. [118]	Manipulation	PPO	2019
Sangiovanni et al. [119]	Obstacle avoidance task of the robotic arm	NAF	2018

of the WECS. Simulation results on a 1.5-MW DFIG wind turbine and experimental tests on an emulated 200 W PMSG (Permanent-Magnet Synchronous Generator) wind turbine confirmed the proposed RL-based MPPT algorithm.

Zhang et al. [103] propose a two-step wind power prediction method, which consists of two phases: long-time-scale coarse prediction and short-time-scale fine correction. In the short-time-scale fine correction, a deep deterministic policy gradient algorithm is implemented to learn the information from real-time weather. The results of the experiments they conducted on the real-life case confirm that their method can properly predict wind power generation and have a better prediction accuracy than existing techniques.

3.2. Electric power systems

Electric power systems face a multitude of control problems over different operating states and time scales where RL could be helpful to handle the problem [103,120–124]. [104,105] solve the classical voltage control problem with the Q-learning algorithm. According to previous research [106–109], automatic generation control of electric power system is achieved by several RL algorithms based on Q-learning. Ye et al. [110] use DDPG algorithms to solve the problem of the market decision of the electric power system.

Zarrabian et al. [111] propose a method based on RL for preventing cascading failure (CF) and blackout in smart grids by acting on the output power of the generators in real-time. The Q-learning algorithm is used in this research to train the system for the optimal action selection strategy. After a period time of training, the system is able to relieve congestion of transmission lines in real-time by adjusting the output power of the generators (actions) to prevent consecutive line outages and blackouts after N-1 and N-1-1 contingency conditions. According to the results of experimental implementation and simulation, their method is accurate and robust in preventing cascading failure and blackout. Moreover, [112] proposes an online Q-learning method to ensure grid security.

All the above, RL implementations in electric power system control and decision problems are mostly based on offline RL [84,86,87,89,91,93,125–130], which is understandable. In electric power systems, the most important issue for designers to be concerned about is safety and reliability. Exploration that is commonly adopted in online RL will be dangerous then. The way to generate a good control policy while ensuring stability is to use offline RL learning with the data previously collected from a system model.

3.3. Coal-fired power plant

DRL has a wide range of applications in traditional systems that use coal to supply energy. In the latest research, Cheng et al. [113] propose a framework called ThermalNet, which uses Long short-term memory (LSTM) components and a DQN network for prediction and optimization, respectively. The system can extract boiler behavior characteristics and formulate a series of control measures, which can reduce emissions and simultaneously enhance fuel utilization according to their experiments.

In a coal-fired power plant system, one of the important tasks is to predict the efficiency of the denitrification and then generate a control strategy to control the efficiency of denitrification based on it to make the power plant's system more efficient. To handle this problem, Fu et al. [114] propose a DRL-based model which combines a LSTM model and A3C algorithm. LSTM is built to give the prediction of the efficiency of the denitrification based on the knowledge learned from the history information. Then the DRL algorithm A3C generates the control strategy for SCR (selective catalytic reduction) denitrification efficiency in coal-fired power plants based on the prediction given by the LSTM. This two-step method is proven to be more accurate and efficient than other machine learning models, as their experiments clearly show. LSTM is efficient in utilizing history information of the power system to make predictions of the future states which can also be introduced to NPPs and leverage temporal knowledge for better system control.

Zhan et al. [115] use a data-driven AI system called DeepThermal to optimize the combustion efficiency of thermal power generating units (TPGUs). The core of this system is a model-based offline RL framework that can solve the MDP problem with pure offline training without real-time interaction with the environment. DeepThermal has been successfully deployed in four large coal-fired thermal power plants in China. Stephan et al. [116] propose an RL system that consists of four agents realized by relatively simple neural function approximators. They demonstrate that the use of this system can significantly reduce air consumption in the production of energy fields.

3.4. Robotics

In general, using a robot to perform certain operations is a relatively complex task, including control and path planning. RL has also been widely and successfully applied in it. Andrychowicz et al. [117] use

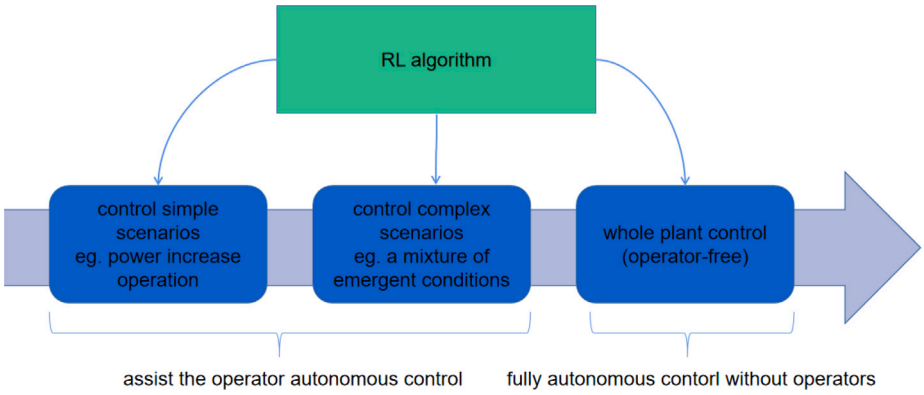

Fig. 5. Roadmap for applying reinforcement learning methods into NPPs.

three technical means, such as extensive randomizations, memory-augmented control policies, and training at large scale with distributed RL, to realize the migration of robots from simulation to real objects. PPO is the most widely used algorithm. By carefully choosing the sensing mode and extensively randomizing the simulation environment, the algorithm can prevent the policy from overfitting a particular simulation scenario, making it more likely to transfer to the real world, and enhancing generalization performance [118]. Aiming at the problem that domain randomization in previous research requires a lot of manual parameter adjustment, a method of automatic domain randomization is proposed, which can automatically complete the parameter adjustment process, thereby further simplifying the real modeling task. In the final experimental environment, the robot can operate the Rubik's cube well.

Sangiovanni [119] faces the obstacle avoidance task of the robotic arm. The reward function is manually designed as the distance between the end effector and the target point, the amplitude of the action, the distance between the obstacle and the robot, and the NAF (normalized advantage function) algorithm is used for training. Among them, the NAF algorithm can be regarded as a continuous version of the Q-learning algorithm. The experiments are carried out in four scenarios, and all have achieved good performance.

4. Reinforcement learning in NPPs

We believe the realization of autonomous and intelligent control of NPPs is the ultimate goal, while there is still a long way to go. We present in Fig. 5 the roadmap of utilizing RL algorithms in NPPs. There are generally three phases:

1. Apply some simple RL algorithms to aid the operators in handling simple tasks in NPPs, e.g., the power increase operation. Such types of tasks are simple and generally can be well-addressed by simple plain control algorithms like PID. Many existing works that attempt to apply RL algorithms in NPPs can be categorized into this type (which is also the key component of the following parts).
2. Apply some advanced and robust RL algorithms to aid the operators in handling some complex tasks in NPPs, e.g., a mixture of emergent conditions occur in NPPs. To the best of our knowledge, this cannot be achieved at the current stage and none of the research succeeds in addressing it. It is hard since the agent needs to quickly identify which types of accidents are happening, and suggest the human operators the correct actions. How to ensure the safety and robustness of the agent are the key challenges.
3. No human operators are needed in the NPPs, and all of the tasks can be done by a single agent. At that time, the whole plant can be all controlled by artificial intelligence, and simultaneously the safety of the NPPs can be guaranteed. This is very challenging, and it needs further advances in the field of artificial intelligence.

At the current stage, we can only realize phase 1 in the roadmap, and we are expecting further advances in this field. Our main contribution in this section is that we provide a block-by-block investigation of the recent advances in applying RL algorithms in NPPs (i.e., papers concerning on phase 1 of the roadmap). We summarize the typical scenarios and algorithms used in Table 2 and hope that our review can aid the researchers of interest and promote the development of nuclear power plant technologies.

4.1. Auto-control and design optimization

4.1.1. Energy system

The energy system can be used to provide the energy supply process of the related equipment in the NPPs, and the setting value of the local controllers needs to be adjusted to perform the corresponding operations. The system is critical for goals such as optimizing heat transfer efficiency, reducing steady-state errors, and more. The use of reinforcement learning control (RLC) can approximate the optimal control capability corresponding to the user-defined utility function, and can effectively avoid the coupling effect and nonlinear relationship between controllers. [131] propose a multi-layer perceptron (MLP)-based state observer and an approximate optimal controller to form a control system. The linear representation of the MLP-based state observer in [131] is given first, and the approximate optimal controller is obtained by solving the Riccati equation corresponding to the linear representation. It can be proven that the closed loop is UUB (Uniformly Ultimately Bounded) stable by the Lyapunov direct method. In specific applications, the MLP-based RLC is applied to the thermal power response optimization of a nuclear steam supply system (NSSS) based on a high-temperature gas-cooled reactor. The simulation results not only show the feasibility and the satisfactory performance of the method but also show the effect of controller parameters on the closed-loop response.

In tackling the control problems associated with boiling water reactors (BWR), characterized by high non-linearity, [133] employs the DDPG algorithm and compares it with the H_∞ control method. The study points out that traditional PID control methods struggle to be effective due to the complexities inherent in BWR control. By utilizing RL-based techniques, these challenges can be well-met. Experiments conducted on a simulator show that the RL control system outperforms the H_∞ control system in areas such as disturbance rejection, perturbation stability, and set-point tracking. The results underscore the suitability of applying RL methods to complex control scenarios like those found in BWR systems.

[132] propose a physics-informed RL optimization method of nuclear assembly design, which could efficiently reduce the cost of energy.

Table 2
An outline of reinforcement learning on NPPs.

NPPs	Operating areas	RL algorithm	Year
Design optimization(Energy system)			
Dong et al. [131]	Nuclear steam supply system	MLP-based RLC	2020
Forgeta et al. [132]	Boiling water reactor	Deep Q-learning, PPO	2021
Chen et al. [133]	Boiling water reactor	DDPG	2022
Seurin et al. [134]	Nuclear fuel loading pattern optimization	PPO	2023
Zhang et al. [135]	Nuclear steam supply system	SAC	2023
Auto-control and monitoring			
Park et al. [136]	Nuclear facilities monitoring	DRL-based SFSC	2020
Lee et al. [137]	Autonomous operation	SAC	2021
Coordinated control			
Li et al. [138]	Coordinated control system	DDPG	2021
Kim et al. [139]	Coordinated control on startup & shutdown part	SAC	2023
Bae et al. [140]	Multi-objective coordinated control	SAC, HER	2023
Operational phase (Power start-up)			
Kim et al. [141]	Power start-up	Q-learning	2019
Lee et al. [142]	Power start-up	A3C	2020
Park et al. [143]	Heat up control	SAC	2022
Operational phase (Emergency operation)			
Lee et al. [137]	Emergency controller	SAC	2021
Application on tokamak: Nuclear fusion			
Degrave et al. [144]	Nuclear fusion controller	MPO	2022

In this study, a connection through reward shaping between RL and the tactics that fuel designers follow in practice has been established, moving fuel rods in the assembly to meet specific constraints and objectives. Their methodology utilizes two classical RL algorithms, DQN, and PPO. They compare the performance of their RL method to SO (stochastic optimization) algorithms such as genetic algorithms and simulated annealing. Applying these methodologies to two boiling water reactor assemblies of low-dimensional (around 2 × 106 combinations) and high-dimensional (around 1031 combinations) natures, they conclude that RL is more effective than SO in solving the high-dimensional problems, through embedding expert knowledge in the form of game rules and effectively exploring the search space. The results of their work demonstrate the effectiveness of RL as another decision-support tool for nuclear fuel assembly optimization.

[134] employs the PPO algorithm for nuclear fuel loading pattern optimization, distinguishing itself from similar studies by deeply experimenting with the effect of different parameters in RL on optimization performance. The study tests two vital parameters, the number of samples seen before updating the model, denoted as n_{steps}, and the parameter controlling the exploration-exploitation trade-off, known as the entropy coefficient, denoted as entropy coefficient. The research identifies the optimal control strategy and parameter range. Specifically, employing a set of specific initial values and reducing n_{steps} and the entropy coefficient until no further learning was observed proved effective. This rigorous examination provides valuable insights and a methodical approach to tuning hyperparameters in the context of nuclear fuel optimization.

NSSS is a crucial component of nuclear power plants, responsible for producing steam used for electricity generation or combined heat and power production. However, the complex coupling between the reactor and steam generator, which cannot be precisely modeled, has limited the performance of existing transient control strategies. [135] introduces a DRL multi-objective optimization method that enhances the responsiveness of both the thermal power and outlet steam temperature within the NSSS by dynamically adjusting the reference values of the existing PID controllers. By merging the strengths of both PID control and DRL methods, this approach exhibits significant improvement in transient response compared to traditional PID controllers. The research presents an innovative solution, offering insights into more effective and responsive control within the complex environment of nuclear power systems.

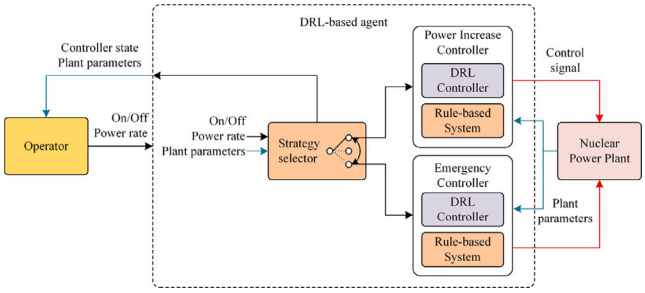

Fig. 6. The structure of DRL-based autonomous agent conducted by Lee et al..

4.1.2. Auto-control and monitoring

Lee et al. [137] suggest a DRL-based autonomous agent. The agent can manage the power increase operation from 2% to 100% and reduce the pressure and temperature until the shutdown cooling entry condition after the reactor trip caused by loss of coolant accident in NPPs. The DRL-based controller suggested in this study combines a rule-based system and DRL algorithm that involves a Soft Actor-Critic (SAC) [82] algorithm and deep neural network. The test results using a compact nuclear simulator indicate that the agent can manipulate components to comply with identified constraints for start-up operation and emergency operation. This study develops a DRL-based agent for power increase and emergency operations. Fig. 6 illustrates the structure of the proposed controller, which consists of (1) a strategy selector, (2) a power increase controller, and (3) an emergency controller.

[136] proposes a data-driven approach using DRL technologies to learn effective strategies to monitor the conditions of nuclear facilities and help the operator diagnose their situations. They design a learning framework and key elements of RL to learn effective strategies and monitor the conditions of nuclear facilities. Based on this, a deep neural network structure and a DRL algorithm are presented for diagnosis learning. Using the scenario data produced from RELAP/MOD3.3 [145], the proposed algorithm has the potential to help in diagnostic tasks. Their experimental results demonstrate the feasibility of DRL in diagnosing the safety functions of a nuclear facility.

4.1.3. Coordinated control

[138] proposes a mathematical model of the coordinated control system, and then transforms it into RL model and applies an existing widely used DRL control algorithm, DDPG to solve the problem.

Through simulation experiments, the proposed algorithm has shown an extremely remarkable control performance. To be specific, the authors came up with a mathematical model to minimize the system deviation, based on the control problem of the reactor-coordinated control system. Considering the uncertainty of the system and the requirements of dynamic real-time optimization, the mathematical model is further transformed into RL control model. Then they utilize the DDPG algorithm to realize interactive learning and problem-solving of the reactor-coordinated control system. In the experimental part, they fully compare the control effects of different DRL algorithms and control schemes formed under random strategies and verify the superior control effect and performance of the DDPG algorithm.

[139] introduces an innovative cooperative control strategy, specifically designed to manage startup and shutdown procedures within nuclear power plants. This strategy subdivides the operations into manageable tasks, each of which is allocated to a designated sub-intelligent agent. These agents employ the SAC algorithm to administer control. Furthermore, to circumvent the occurrence of mutually conflicting actions among the diverse agents, the manuscript delineates the design of a superior decision-making system. This system exploits a LSTM network, which is used to anticipate potential future parameters of the nuclear power plant's operations. By employing a scoring mechanism that selects the most appropriate action based on the highest score, this framework facilitates harmonious and coordinated actions amongst the various intelligent agents. This represents a significant stride in enhancing operational efficiency and mitigating potential conflicts in nuclear power plant management.

Addressing coordinated control problems by achieving multiple objective multi-objective control problems, [140] introduces a novel scheme employing SAC and Hindsight Experience Replay (HER) algorithms to manage continuous control tasks under a simple binary reward function. Experiments were set up involving five agents, tasked with individually controlling the pressure and volume of reactor coolant during startup. The results yield favorable performance, even revealing that the DRL method can achieve good outcomes on untrained objectives, such as the cooling task of the coolant.

4.2. Operational phase

4.2.1. Power start-up

The power start-up of the NPPs is a very critical task and the goal is to automatically increase the power of the reactor from around 2% to 100% so that the subsequent temperature increases and other complete start-up processes can be carried out. Generally speaking, a nuclear power plant should be a system with a high level of intelligence, because the tasks between the various subsystems must be executed quickly, and any command needs to be responded swiftly to ensure overall safety. However, traditional power boosting is operated manually, which is more prone to operational errors due to the following reasons. The number of target selections and strategies to be adopted is rapidly increasing. There are a large number of related parameters that need to be additionally maintained, tested, and detected. Meanwhile, the operator may not have clear operating instructions [146–149]. Using RL as an auxiliary operating system to replace traditional manual operations can effectively reduce operational errors on this task, reduce operating costs, and improve energy utilization.

Lee et al. [142] propose a DRL framework based on the A3C algorithm while incorporating the LSTM network for continuous control, and discrete control is performed using rule-based design operations. Both of them are combined as the full process of power activation. Since the process of power startup involves a series of timing operations, the LSTM block is introduced. This network is a variant of the RNN network, which can process sequential signals well. As a typical algorithm in DRL, the A3C algorithm can effectively improve CPU utilization, reduce the number of interactions with the environment, and speed up training efficiency. In this task, the reward function is set as the deviation of the current power and the expected power at the moment, and the paper sets an upper and lower bound for the power situation at different moments to maintain the reward that should be obtained. If at some point the power is outside this limit, the training stops. Finally, the continuous modules that need to be controlled will be output, such as rod controller, make-up water valve and boric acid water valve, and other components that should take action indicators. Experiments show that the algorithm can even increase the power from 2% to 100% according to the degree of 3%/h. Besides, [141] has come up with a two-level method using Q-learning to overcome the difficulty of lacking data on the start-up and shut-down part. The first level aims at outputting goals and constraints with a supervisory operation module. The second level accepts the outputs of the first level and attempts to achieve the desired goal under certain constraints using either an RL algorithm or rule-based control. The Q-learning algorithm is adopted to solve tasks that require complex judgment and component control, and otherwise, simple rule-based logic is utilized for task solving.

[143] addresses the application of DRL to the heat-up issues in nuclear power plants by proposing the use of the A3C algorithm for control. Additionally, a neural network framework structure was designed to facilitate the synchronization of information among different agents. Furthermore, the paper expanded the existing compact nuclear simulator (CNS) interface to enable compatibility with custom RL frameworks and algorithms.

4.2.2. Emergency operation

Under certain circumstances, NPPs may produce unexpected accidents. At this time, we can adopt a series of operational measures to ensure the integrity of the core sector and prevent the risk of major problems to the greatest extent. The technical means used in traditional non-automatic NPPs are manual intervention. That is, to predict in time before the accident, activate the safety system in time, and manually cool the nuclear reactor. However such a method is not automated enough and could not function in time. [137] presents a state-of-the-art SAC based algorithm that can choose suitable actions to meet pressure and temperature demand profiles, as well as to achieve a specified cooling rate.

The controller proposed in the paper is generally composed of three parts: a policy selector, a power increase controller, and an emergency controller, which respectively apply a rule-based system for discrete control and an SAC algorithm from RL for continuous control. SAC has been proven to be efficient at exploring various continuous control tasks [82] and can avoid those unnecessary branches that could bother the training procedure. This algorithm generally is applied in continuous control, which demands higher accuracy while prior knowledge can make fewer contributions. The control target is those parameters of the PZR (the pressure of pressurizer) spray valve, aux feed-water valve, and steam dump valve, which have significant relationships with the two key factors determining the resilience when meeting the constraints of pressure and temperature. The reward is designed to be the change in temperature and pressure. In other words, the less of the change in those two values, the larger of reward the agent could receive.

The authors' team utilized a CNS as a real-time testbed for training and validating the proposed algorithm. The emergency condition adopted in the paper is the Loss of Coolant Accidents (LOCA). Experiments show that after training for a long enough period, the RL algorithm proposed in this paper can effectively cool the reactor temperature and satisfy various constraints. Experiments indicate that this algorithm can also reach the shutdown operation entry condition.

4.3. Application on tokamak: Nuclear fusion

Recently, Degrave et al. [144] publish an article in Nature on nuclear fusion using DRL, which has attracted widespread attention. This work has laid a solid foundation for large-scale deployment in NPPs in the future to achieve sustainable energy. As a tokamak, the

Fig. 7. Around the world, there are several organizations and nations that are dedicated to RL's participation in the NPPs. Among them are South Korea, China, the United States, Australia, and the United Kingdom. It is important to note that Korea leads the pack in terms of outcomes, accounting for nearly half of the production. On the other hand, related research is far less developed in Europe and North America, and it is hoped that these areas could pick up the pace and actively support the growth of this sector.

hydrogen atoms inside a ring-shaped container that can hold a nuclear fusion reaction will generate a rotating and tumbling plasma, which makes the tokamak appear in a chaotic state. If the plasma can be controlled so that it is stable enough, it could be a big step forward in achieving controlled nuclear fusion. The difficulty of the problem is not only continuously maintaining the high temperature of the plasma in the container but also achieving different complex configurations for different plasmas. This is a time-varying, multi-variable and nonlinear control task, and the authors utilize RL to carry out it.

The difficulty of tokamak nuclear fusion is to maintain the continuous high temperature of the plasma in the container, which requires high-frequency and continuous control of the magnetically actuated coil. There are also different control objectives at different stages. In this paper, a new controller design architecture is proposed. The model is divided into three stages. The first stage is to design the desired goal of the designer for the experiment. Goals may include the location of the plasma and the stability of its current flow. Considering the different configurations required by the plasma in different states, this goal may change over time. The second stage is to use DRL to interact with the tokamak simulator to find a near-optimal policy that can achieve the goal. Finally, in the third stage, the control policy is deployed to the tokamak hardware for operation in real life. Based on the free boundary plasma evolution model, the plasma state is modeled under the influence of the coil voltage. To cope with the mismatch of the corresponding data rate between the simulator and the RL environment, the researchers use MPO algorithm.

In the experimental session, the author's team first demonstrated the precise control of the fundamental quality of the plasma equilibrium. The results show that the RL architecture enables precise plasma control in all relevant phases of the discharge experiment. Furthermore, there is much evidence showing that the proposed architecture is capable of scientific research to generate complex configurations. Finally, the authors test the control of "droplets", a configuration in which two independent plasmas coexist inside the vessel, to demonstrate the power of the architecture in exploring new plasma configurations. This work lays an important foundation for the combination of basic disciplines and RL and the subsequent application of RL in nuclear fission as well as in NPPs.

5. Discussion

In this section, we provide some detailed discussions and suggestions to aid the development of incorporating RL algorithms with NPPs.

5.1. What can we learn from the collected papers?

Based on our review, we first plot the word cloud diagram on key nations, universities, or institutions that apply RL algorithms to NPPs, and the key RL algorithms existing literature employed in NPPs, which can be found in Fig. 7. We find that in terms of nations, China and South Korea publish the most articles on utilizing RL algorithms in NPPs; in terms of the institutions, Chosun University and Tsinghua University are the most energetic in exploring the possibilities of RL algorithms in NPPs; as for the RL algorithms, we can see that DQN and SAC are the most popular algorithms adopted. The diagram results indicate that the Asian countries and institutions are the most positive in introducing recent advances in the RL community into the scenario of NPPs, and they tend to use the widely investigated algorithms instead of a new algorithm that has not been tested in domains other than simulated simple control tasks like locomotion. It is understandable since the simulations of NPPs are time-consuming, the real-world application in NPPs is expensive and risky, and there is no guarantee that new techniques can incur a good result. For the benefit of developing the next generation advanced NPPs, we list the following suggestions in the research community of nuclear energy and nuclear power plants: (1) countries or institutions outside Asia ought to engage more into this direction and work together to foster the promising opportunity of combining RL algorithms into NPPs; (2) the community ought to actively try the possibility of more advanced RL algorithms in different NPP tasks, e.g., model-based RL [93,94,150–152], offline RL [85–87,153,154], etc.

5.2. Why should we introduce RL algorithms into NPPs?

We can see that, in a series of tasks in NPPs, process control tasks occupy the mainstream. Most of the controllers in these tasks are generated based on models, so we need to maintain and update the models regularly to ensure their performance. Andersen [155] and Spielberg [156] have conducted relevant theoretical research on the application of DRL in process control and verified its effectiveness and superiority through a large number of simulations. Traditional PID controllers require continuous monitoring and call remedial models when their performance degrades. Once a failure occurs, the maintenance process used is usually complex and can cause work interruptions. In some complex application scenarios of NPPs, when faced with nonlinear and high-dimensional control tasks, it is difficult to model high-quality models, and it is challenging to ensure the effectiveness of PID controllers. The DRL controller can skip the process of artificially modeling the environment by learning in the process of interacting with the environment, i.e., the NPP systems. Generally speaking, the RL algorithms have a more specific understanding of the environment

Table 3
List of abbreviations and their full forms in this paper.

Abbreviations	Full forms
NPP	Nuclear Power Plants
RL	Reinforcement Learning
PID	Proportional-integral-differential
PLCs	Programmable logic controllers
FPGAs	Field-programmable gate arrays
DRL	Deep reinforcement learning
AI	Artificial intelligence
MDP	Markov decision process
DQN	Deep Q-Network
TRPO	Trust Region Policy Optimization
DDPG	Deep Deterministic Policy Gradient
A3C	Asynchronous Advantage Actor-Critic
VPG	Vanilla Policy Gradient
PPO	Proximal Policy Optimization
MASs	Multiagent systems
ESS	Energy Storage System
WPG	Wind Power Generation
ES	Energy subsystem
PS	Power subsystem
MAE	Mean Absolute Error
MPPT	Maximum power point tracking
WECSs	Wind energy conversion
MPPS	Maximum power points
PMSG	Permanent-Magnet Synchronous Generator
AGC	Automatic Generation Contro
LSTM	Long short-term memory
NSSS	Nuclear steam supply system
SO	Stochastic optimization
SAC	Soft Actor-Critic
HER	Hindsight Experience Replay
CNS	Compact nuclear simulator
LOCA	Loss of coolant accidents

and can achieve some intelligence instead of controlling by following a previously defined rule. DRL methods have more flexibility compared to PID controllers.

A recent work [157] also indicates that, compared to employing traditional PID controllers, the utilization of DRL controllers results in smaller accumulative error. Moreover, combining the two by using a DRL-tuned PID controller generates the smallest error and quicker response times. Furthermore, while PID-based controllers tend to focus on individual segments, DRL-based controllers can simultaneously control multiple parts and sets of parameters. Finally, in contrast to PID controllers, DRL controllers feature lower operation frequency, which leads to two main advantages: reduction in the probability of unsuccessful operations, and prevention of premature aging in instruments and equipment.

5.3. The benefits and problems of using RL algorithms in NPPs

By observing the various applications of RL in complex systems, it is not difficult to find that, relying on the powerful tool of the deep neural network, RL has gradually replaced traditional control methods and plays an increasingly important role in many control tasks. Using DRL algorithms can help us achieve good fitting results in many nonlinear complex environments without relying on too much expert knowledge. In addition, we can also achieve end-to-end control with the help of neural networks, thus avoiding the intermediate data conversion process.

By investigating the application of existing RL in NPPs, we can see that some of the work has achieved good performance in various modules of NPPs. However, it is undeniable that the related work is comparatively blank and fragmented than other fields like robotics, game AI, etc. It can be seen that DRL is easily applied to tasks based on continuous control, and is matched with discrete control based on prior knowledge to complete control tasks together. In addition, these RL algorithms are basically model-free, and another mainstream direction in RL, model-based methods, is rarely studied, which shows that there are broad spaces on the application of model-based methods on NPPs.

However, there are also some problems in applying DRL methods in NPPs which we summarize below. We give some current efforts and directions for addressing these challenges.

- **The sampling efficiency of RL algorithms is usually poor.** To address the poor sample efficiency issue in online RL algorithms [47,82,158,159], many types of research on applying RL in NPPs use distributed training models, which can ensure full utilization of data and improve training efficiency for subsequent deployment to large-scale complex scenarios. Considering that control tasks often need to combine historical data, LSTM networks are also widely used. These all can remedy the potential issues of vanilla RL algorithms and contribute to a better-behaving agent in NPPs.
- **There is a gap between simulation and real-life operation.** At present, most of the advances in applying DRL algorithms in NPPs are evaluated in simulators. However, there is a natural gap between the simulator and the real-world NPP system. The error between simulation and reality can potentially incur several drops in performance when migrating the well-behaved agent that is achieved in simulation and small-scale experimental conditions to large-scale or real environments. It tends to be frustrating that the same excellent performance cannot be guaranteed. In recent years, scholars in the field of sim-to-real at NPPs have introduced the concept of *Robust AI* [160,161]. This approach focuses on utilizing meta-knowledge inherent in data, rather than the data itself, for training purposes. To a certain extent, this mitigates the inconsistencies between simulations and real-world scenarios. There are many other efforts in mitigating this gap, namely sim-to-real for robotics and some other situations [162–169], and these advances can be hopefully applied in NPPs in future work to acquire a robust and reliable agent.
- **The interpretability of the model is inadequate.** When we use RL to assist humans in some tasks in NPPs, it is necessary to pay close attention to the training process and the result of the RL agent. DRL itself is a black-box nature. If there is a problem of poor performance, we cannot hope that the relevant personnel has rich experience and repair methods as in traditional control. Therefore, in the current stage, we cannot expect that the RL agent can completely replace human operators. They can be utilized as an assistant during operations. Moreover, there is much progress in the interpretability of RL and deep neural networks [170–172].
- **The generalization of the model needs to be further enhanced.** The generalization capability of the DRL agent is of importance to the RL community and there are many types of research on this topic [173–177]. We find that all of our reviewed papers set their focus on a small part of the whole NPP system, such as a single heater or power starter. There is still a broad space for exploration in the application of DRL in NPPs. For example, can DRL successfully control multiple devices or even the whole plant under some situations, e.g., emergence operation? Can we train a single DRL agent that can solve many different tasks in NPPs with a few interactions? Can we train a DRL agent on one single task and then transfer it to a different task in NPPs? These questions are all valuable and are expected to be solved in the near future.

Despite the aforementioned issues of utilizing DRL methods in NPPs, we believe it is very promising to combine DRL methods with NPPs. These challenges also bring about many open problems for researchers to address. We think the utilization of DRL algorithms in NPPs is one of the most possible to realize the intelligent unmanned NPPs.

6. Conclusion

This article focuses on the application of reinforcement learning in the field of artificial intelligence in NPPs. We first introduce some mainstream algorithms in RL, most of which have been applied to specific NPP scenarios in some studies. Afterward, considering the complex system nature of NPPs, we investigate the application of RL in the power grid, wind power, thermal energy, robotics, etc., to pave the way for subsequent use in NPPs. Finally, we conduct a block-by-block investigation on the application of RL methods in some specific tasks in NPPs by extensively reviewing the current research and advances of the combination of DRL algorithms and NPPs. We carry out algorithmic research for different situations such as power startup, collaborative control, and emergency handling.

Unfortunately, there are quite a few articles about the application of RL in NPPs. This field is still comparatively blank and many works can be explored. Given the fact of the extensive application of RL in complex systems, and the unique advantages of RL in dealing with nonlinear complex problems, we expect that more advances in applying DRL algorithms in NPPs can receive more attention and have further development.

CRediT authorship contribution statement

Aicheng Gong: Conceptualization, Methodology, Writing. **Yangkun Chen:** Methodology, Writing. **Junjie Zhang:** Data curation, Writing. **Xiu Li:** Supervision.

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.

References

[1] Vivianne H.M. Visschers, Carmen Keller, Michael Siegrist, Climate change benefits and energy supply benefits as determinants of acceptance of nuclear power stations: Investigating an explanatory model, Energy Policy 39 (6) (2011) 3621–3629.

[2] Sheng Zhou, Xiliang Zhang, Nuclear energy development in China: a study of opportunities and challenges, Energy 35 (11) (2010) 4282–4288.

[3] Barry W Brook, Agustin Alonso, Daniel A Meneley, Jozef Misak, Tom Blees, Jan B van Erp, Why nuclear energy is sustainable and has to be part of the energy mix, Sustain. Mater. Technol. 1 (2014) 8–16.

[4] James H. Rust, Nuclear power plant engineering, 1979.

[5] Ronald Allen Knief, Nuclear energy technology: theory and practice of commercial nuclear power, 1981.

[6] José M. Arias, Manuel Lozano, An Advanced Course in Modern Nuclear Physics, Vol. 581, Springer, 2008.

[7] Mingrong Li Tingke Zhang, The Report On The Development of China's Nuclear Energy 2021, Technical Report, EG and G Idaho, Inc., Idaho Falls (USA), 2021.

[8] Steffen Schlömer, Thomas Bruckner, Lew Fulton, Edgar Hertwich, Alan McKinnon, Daniel Perczyk, Joyashree Roy, Roberto Schaeffer, Ralph Sims, Pete Smith, et al., Annex III: Technology-specific cost and performance parameters, in: Climate Change 2014: Mitigation of Climate Change: Contribution of Working Group III to the Fifth Assessment Report of the Intergovernmental Panel on Climate Change, Cambridge University Press, 2014, pp. 1329–1356.

[9] Henry P. Birmingham, Franklin V. Taylor, A design philosophy for man-machine control systems, Proc. IRE 42 (12) (1954) 1748–1758.

[10] D.T. McRuer, E.S. Krendel, The man-machine system concept, Proc. IRE 50 (5) (1962) 1117–1123.

[11] Jean-Michel Hoc, From human–machine interaction to human–machine cooperation, Ergonomics 43 (7) (2000) 833–843.

[12] Hang Wu, Weihua Su, Zhiguo Liu, PID controllers: Design and tuning methods, in: 2014 9th IEEE Conference on Industrial Electronics and Applications, IEEE, 2014, pp. 808–813.

[13] Rames C. Panda, Introduction to PID Controllers: Theory, Tuning and Application to Frontier Areas, BoD–Books on Demand, 2012.

[14] MA Ebrahim, KA El-Metwally, FM Bendary, WM Mansour, HS Ramadan, R Ortega, J Romero, Optimization of proportional-integral-differential controller for wind power plant using particle swarm optimization technique, Int. J. Electr. Power Eng. 6 (1) (2012) 32–37.

[15] Lezhen Shi, Xiaodong Miao, Hua Wang, An improved nonlinear proportional-integral-differential controller combined with fractional operator and symbolic adaptation algorithm, Trans. Inst. Meas. Control 42 (5) (2020) 927–941.

[16] Stuart Bennett, Development of the PID controller, IEEE Control Syst. Mag. 13 (6) (1993) 58–62.

[17] Kit-Sang Tang, Kim Fung Man, Guanrong Chen, Sam Kwong, An optimal fuzzy PID controller, IEEE Trans. Ind. Electron. 48 (4) (2001) 757–765.

[18] Pritesh Shah, Sudhir Agashe, Review of fractional PID controller, Mechatronics 38 (2016) 29–41.

[19] Kelvin T. Erickson, Programmable logic controllers, IEEE Potentials 15 (1) (1996) 14–17.

[20] William Bolton, Programmable Logic Controllers, Newnes, 2015.

[21] Ephrem Ryan Alphonsus, Mohammad Omar Abdullah, A review on the applications of programmable logic controllers (PLCs), Renew. Sustain. Energy Rev. 60 (2016) 1185–1205.

[22] Gary A. Dunning, Introduction to Programmable Logic Controllers, Cengage Learning, 2005.

[23] Eric Monmasson, Marcian N. Cirstea, FPGA design methodology for industrial control systems—A review, IEEE Trans. Ind. Electron. 54 (4) (2007) 1824–1842.

[24] Ian Kuon, Russell Tessier, Jonathan Rose, et al., FPGA architecture: Survey and challenges, Found. Trends® Electron. Des. Autom. 2 (2) (2008) 135–253.

[25] Andrew Canis, Jongsok Choi, Mark Aldham, Victor Zhang, Ahmed Kammoona, Jason H Anderson, Stephen Brown, Tomasz Czajkowski, LegUp: high-level synthesis for FPGA-based processor/accelerator systems, in: Proceedings of the 19th ACM/SIGDA International Symposium on Field Programmable Gate Arrays, 2011, pp. 33–36.

[26] Teng Lei, Wang Shuai, Wang Xiaobing, Design of high pressure water jet decontamination device with pressure and flow synchronous control function, Nucl. Power Eng. 41 (3) (2020) 153–157.

[27] Jonghyun Kim, Seungjun Lee, Poong Hyun Seong, Autonomous Nuclear Power Plants with Artificial Intelligence, Vol. 94, Springer Nature, 2023.

[28] Jürgen Schmidhuber, Deep learning in neural networks: An overview, Neural Netw. 61 (2015) 85–117.

[29] Yann LeCun, Yoshua Bengio, Geoffrey Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[30] Li Deng, Dong Yu, et al., Deep learning: methods and applications, Found. Trends® Signal Process. 7 (3–4) (2014) 197–387.

[31] Pramila P. Shinde, Seema Shah, A review of machine learning and deep learning applications, in: 2018 Fourth International Conference on Computing Communication Control and Automation (ICCUBEA), IEEE, 2018, pp. 1–6.

[32] Ajeet Ram Pathak, Manjusha Pandey, Siddharth Rautaray, Application of deep learning for object detection, Procedia Comput. Sci. 132 (2018) 1706–1717.

[33] Zongmei Gao, Zhongwei Luo, Wen Zhang, Zhenzhen Lv, Yanlei Xu, Deep learning application in plant stress imaging: a review, AgriEngineering 2 (3) (2020) 29.

[34] Samir Khan, Takehisa Yairi, A review on the application of deep learning in system health management, Mech. Syst. Signal Process. 107 (2018) 241–265.

[35] Richard S. Sutton, Andrew G. Barto, Reinforcement Learning: An Introduction, MIT Press, 2018.

[36] Leslie Pack Kaelbling, Michael L. Littman, Andrew W. Moore, Reinforcement learning: A survey, J. Artif. Intell. Res. 4 (1996) 237–285.

[37] Yuxi Li, Deep reinforcement learning: An overview, 2017, arXiv preprint arXiv: 1701.07274.

[38] James Ladyman, James Lambert, Karoline Wiesner, What is a complex system? Eur. J. Philos. Sci. 3 (1) (2013) 33–67.

[39] Takayuki Kanda, Hiroshi Ishiguro, Michita Imai, Tetsuo Ono, Development and evaluation of interactive humanoid robots, Proc. IEEE 92 (11) (2004) 1839–1850.

[40] Bradley T. Werner, Complexity in natural landform patterns, Science 284 (5411) (1999) 102–104.

[41] Klaus Mainzer, A. John Mallinckrodt, Thinking in complexity: The complex dynamics of matter, mind, and mankind, Comput. Phys. 9 (4) (1995) 398.

[42] Robert S. MacKay, Nonlinearity in complexity science, Nonlinearity 21 (12) (2008) T273.

[43] Alicia Juarrero, Dynamics in action: Intentional behavior as a complex system, Emergence 2 (2) (2000) 24–57.

[44] Sven Bertelsen, Construction as a complex system, in: Proceedings for the 11th Annual Conference of the International Group for Lean Construction, 2003, pp. 11–23.

[45] Steven D. Gribble, Robustness in complex systems, in: Proceedings Eighth Workshop on Hot Topics in Operating Systems, IEEE, 2001, pp. 21–26.

[46] Herbert A. Simon, The architecture of complexity, in: Facets of Systems Science, Springer, 1991, pp. 457–476.

[47] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al., Human-level control through deep reinforcement learning, Nature 518 (7540) (2015) 529–533.

[48] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, et al., Mastering the game of go without human knowledge, Nature 550 (7676) (2017) 354–359.

[49] Timothy P Lillicrap, Jonathan J Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, Daan Wierstra, Continuous control with deep reinforcement learning, 2015, arXiv preprint arXiv:1509.02971.

[50] Sergey Levine, Chelsea Finn, Trevor Darrell, Pieter Abbeel, End-to-end training of deep visuomotor policies, J. Mach. Learn. Res. 17 (1) (2016) 1334–1373.

[51] Oriol Vinyals, Igor Babuschkin, Wojciech M Czarnecki, Michaël Mathieu, Andrew Dudzik, Junyoung Chung, David H Choi, Richard Powell, Timo Ewalds, Petko Georgiev, et al., Grandmaster level in StarCraft II using multi-agent reinforcement learning, Nature 575 (7782) (2019) 350–354.

[52] Julien Perolat, Bart de Vylder, Daniel Hennes, Eugene Tarassov, Florian Strub, Vincent de Boer, Paul Muller, Jerome T Connor, Neil Burch, Thomas Anthony, et al., Mastering the game of stratego with model-free multiagent reinforcement learning, 2022, arXiv e-prints, arXiv–2206.

[53] Jens Kober, J. Andrew Bagnell, Jan Peters, Reinforcement learning in robotics: A survey, Int. J. Robot. Res. 32 (11) (2013) 1238–1274.

[54] Petar Kormushev, Sylvain Calinon, Darwin G. Caldwell, Reinforcement learning in robotics: Applications and real-world challenges, Robotics 2 (3) (2013) 122–148.

[55] Athanasios S. Polydoros, Lazaros Nalpantidis, Survey of model-based reinforcement learning: Applications on robotics, J. Intell. Robot. Syst. 86 (2) (2017) 153–173.

[56] Wenshuai Zhao, Jorge Peña Queralta, Tomi Westerlund, Sim-to-real transfer in deep reinforcement learning for robotics: a survey, in: 2020 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE, 2020, pp. 737–744.

[57] Zhong-Qiu Zhao, Peng Zheng, Shou-tao Xu, Xindong Wu, Object detection with deep learning: A review, IEEE Trans. Neural Netw. Learn. Syst. 30 (11) (2019) 3212–3232.

[58] Md Tahmid Hasan Fuad, Awal Ahmed Fime, Delowar Sikder, Md Akil Raihan Iftee, Jakaria Rabbi, Mabrook S Al-Rakhami, Abdu Gumaei, Ovishake Sen, Mohtasim Fuad, Md Nazrul Islam, Recent advances in deep learning techniques for face recognition, IEEE Access 9 (2021) 99112–99142.

[59] Dinggang Shen, Guorong Wu, Heung-Il Suk, Deep learning in medical image analysis, Annu. Rev. Biomed. Eng. 19 (2017) 221.

[60] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud Arindra Adiyoso Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen Awm Van Der Laak, Bram Van Ginneken, Clara I Sánchez, A survey on deep learning in medical image analysis, Med. Image Anal. 42 (2017) 60–88.

[61] Afan Galih Salman, Bayu Kanigoro, Yaya Heryadi, Weather forecasting using deep learning techniques, in: 2015 International Conference on Advanced Computer Science and Information Systems (ICACSIS), IEEE, 2015, pp. 281–285.

[62] Tom Young, Devamanyu Hazarika, Soujanya Poria, Erik Cambria, Recent trends in deep learning based natural language processing, IEEE Comput. Intell. Mag. 13 (3) (2018) 55–75.

[63] Yushi Chen, Zhouhan Lin, Xing Zhao, Gang Wang, Yanfeng Gu, Deep learning-based classification of hyperspectral data, IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens. 7 (6) (2014) 2094–2107.

[64] Zidong Zhang, Dongxia Zhang, Robert C. Qiu, Deep reinforcement learning for power system applications: An overview, CSEE J. Power Energy Syst. 6 (1) (2019) 213–225.

[65] Nguyen Cong Luong, Dinh Thai Hoang, Shimin Gong, Dusit Niyato, Ping Wang, Ying-Chang Liang, Dong In Kim, Applications of deep reinforcement learning in communications and networking: A survey, IEEE Commun. Surv. Tutor. 21 (4) (2019) 3133–3174.

[66] Hao-nan Wang, Ning Liu, Yi-yun Zhang, Da-wei Feng, Feng Huang, Dong-sheng Li, Yi-ming Zhang, Deep reinforcement learning: a survey, Front. Inf. Technol. Electron. Eng. 21 (12) (2020) 1726–1744.

[67] Amirhosein Mosavi, Yaser Faghan, Pedram Ghamisi, Puhong Duan, Sina Faizollahzadeh Ardabili, Ely Salwana, Shahab S Band, Comprehensive review of deep reinforcement learning methods and applications in economics, Mathematics 8 (10) (2020) 1640.

[68] Ammar Haydari, Yasin Yilmaz, Deep reinforcement learning for intelligent transportation systems: A survey, IEEE Trans. Intell. Transp. Syst. (2020).

[69] Christopher John Cornish Hellaby Watkins, Learning from delayed rewards, 1989.

[70] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, Martin Riedmiller, Playing atari with deep reinforcement learning, 2013, arXiv preprint arXiv:1312.5602.

[71] Leemon Baird, Residual algorithms: Reinforcement learning with function approximation, in: Machine Learning Proceedings 1995, Elsevier, 1995, pp. 30–37.

[72] John Tsitsiklis, Benjamin Van Roy, Analysis of temporal-diffference learning with function approximation, Adv. Neural Inf. Process. Syst. 9 (1996).

[73] Long-Ji Lin, Reinforcement Learning for Robots using Neural Networks, Carnegie Mellon University, 1992.

[74] Hado Hasselt, Double Q-learning, Adv. Neural Inf. Process. Syst. 23 (2010).

[75] Hado Van Hasselt, Arthur Guez, David Silver, Deep reinforcement learning with double q-learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, Vol. 30, 2016.

[76] Richard S Sutton, David McAllester, Satinder Singh, Yishay Mansour, Policy gradient methods for reinforcement learning with function approximation, Adv. Neural Inf. Process. Syst. 12 (1999).

[77] John Schulman, Philipp Moritz, Sergey Levine, Michael Jordan, Pieter Abbeel, High-dimensional continuous control using generalized advantage estimation, 2015, arXiv preprint arXiv:1506.02438.

[78] Ivaylo Popov, Nicolas Heess, Timothy Lillicrap, Roland Hafner, Gabriel Barth-Maron, Matej Vecerik, Thomas Lampe, Yuval Tassa, Tom Erez, Martin Riedmiller, Data-efficient deep reinforcement learning for dexterous manipulation, 2017, arXiv preprint arXiv:1704.03073.

[79] John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, Philipp Moritz, Trust region policy optimization, in: International Conference on Machine Learning, PMLR, 2015, pp. 1889–1897.

[80] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, Oleg Klimov, Proximal policy optimization algorithms, 2017, arXiv preprint arXiv:1707.06347.

[81] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, Koray Kavukcuoglu, Asynchronous methods for deep reinforcement learning, in: International Conference on Machine Learning, PMLR, 2016, pp. 1928–1937.

[82] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, Sergey Levine, Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor, in: International Conference on Machine Learning, PMLR, 2018, pp. 1861–1870.

[83] Abbas Abdolmaleki, Jost Tobias Springenberg, Yuval Tassa, Remi Munos, Nicolas Heess, Martin Riedmiller, Maximum a posteriori policy optimisation, 2018, arXiv preprint arXiv:1806.06920.

[84] Scott Fujimoto, David Meger, Doina Precup, Off-policy deep reinforcement learning without exploration, in: International Conference on Machine Learning, 2019.

[85] Aviral Kumar, Aurick Zhou, George Tucker, Sergey Levine, Conservative q-learning for offline reinforcement learning, Adv. Neural Inf. Process. Syst. 33 (2020) 1179–1191.

[86] Jiafei Lyu, Xiaoteng Ma, Xiu Li, Zongqing Lu, Mildly conservative Q-learning for offline reinforcement learning, in: Thirty-Sixth Conference on Neural Information Processing Systems, 2022.

[87] Scott Fujimoto, Shixiang Shane Gu, A minimalist approach to offline reinforcement learning, in: Advances in Neural Information Processing Systems, 2021.

[88] Aviral Kumar, Justin Fu, G. Tucker, Sergey Levine, Stabilizing off-policy Q-learning via bootstrapping error reduction, in: Neural Information Processing Systems, 2019.

[89] Jiafei Lyu, aicheng Gong, Le Wan, Zongqing Lu, Xiu Li, State advantage weighting for offline RL, in: 3rd Offline RL Workshop: Offline RL As a ”Launchpad”, 2022.

[90] Michael Janner, Justin Fu, Marvin Zhang, Sergey Levine, When to trust your model: Model-based policy optimization, 2019, ArXiv, abs/1906.08253.

[91] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Y. Zou, Sergey Levine, Chelsea Finn, Tengyu Ma, MOPO: Model-based offline policy optimization, in: Advances in Neural Information Processing Systems, 2020.

[92] Tianhe Yu, Aviral Kumar, Rafael Rafailov, Aravind Rajeswaran, Sergey Levine, Chelsea Finn, COMBO: Conservative offline model-based policy optimization, in: Neural Information Processing Systems, 2021.

[93] Jiafei Lyu, Xiu Li, Zongqing Lu, Double check your state before trusting it: Confidence-aware bidirectional offline model-based imagination, 2022, arXiv preprint arXiv:2206.07989.

[94] Junjie Zhang, Jiafei Lyu, Xiaoteng Ma, Jiangpeng Yan, Jun Yang, Le Wan, Xiu Li, Uncertainty-driven trajectory truncation for model-based offline reinforcement learning, 2023, ArXiv, abs/2304.04660.

[95] Zhongjian Qiao, Jiafei Lyu, Xiu Li, The primacy bias in model-based RL, 2023, ArXiv, abs/2310.15017.

[96] Marc Rigter, Bruno Lacerda, Nick Hawes, RAMBO-RL: Robust adversarial model-based offline reinforcement learning, 2022, ArXiv, abs/2204.12581.

[97] Suyang Zhou, Zijian Hu, Wei Gu, Meng Jiang, Meng Chen, Qiteng Hong, Campbell Booth, Combined heat and power system intelligent economic dispatch: A deep reinforcement learning approach, Int. J. Electr. Power Energy Syst. 120 (2020) 106016.

[98] Esmat Samadi, Ali Badri, Reza Ebrahimpour, Decentralized multi-agent based energy management of microgrid using reinforcement learning, Int. J. Electr. Power Energy Syst. 122 (2020) 106211.

[99] Hussain Kazmi, Johan Suykens, Attila Balint, Johan Driesen, Multi-agent reinforcement learning for modeling and control of thermostatically controlled loads, Appl. Energy 238 (2019) 1022–1035.

[100] Tianshu Wei, Yanzhi Wang, Qi Zhu, Deep reinforcement learning for building HVAC control, in: Proceedings of the 54th Annual Design Automation Conference 2017, 2017, pp. 1–6.

[101] Eunsung Oh, Hanho Wang, Reinforcement-learning-based energy storage system operation strategies to manage wind power forecast uncertainty, IEEE Access 8 (2020) 20965–20976.

[102] Chun Wei, Zhe Zhang, Wei Qiao, Liyan Qu, Reinforcement-learning-based intelligent maximum power point tracking control for wind energy conversion systems, IEEE Trans. Ind. Electron. 62 (10) (2015) 6360–6370.
[103] Huifeng Zhang, Dong Yue, Chunxia Dou, Kang Li, Gerhard P Hancke, Two-step wind power prediction approach with improved complementary ensemble empirical mode decomposition and reinforcement learning, IEEE Syst. J. (2021).
[104] Yinliang Xu, Wei Zhang, Wenxin Liu, Frank Ferrese, Multiagent-based reinforcement learning for optimal reactive power dispatch, IEEE Trans. Syst. Man Cybern. C 42 (6) (2012) 1742–1751.
[105] John G. Vlachogiannis, Nikos D. Hatziargyriou, Reinforcement learning for reactive power control, IEEE Trans. Power Syst. 19 (3) (2004) 1317–1325.
[106] Tao Yu, Bin Zhou, Ka Wing Chan, Liang Chen, Bo Yang, Stochastic optimal relaxed automatic generation control in non-Markov environment based on multi-step q learning, IEEE Trans. Power Syst. 26 (3) (2011) 1272–1282.
[107] Fatheme Daneshfar, Hassan Bevrani, Load–frequency control: a GA-based multi-agent reinforcement learning, IET Gener. Transm. Distrib. 4 (1) (2010) 13–26.
[108] T.P. Imthias Ahamed, P.S. Nagendra Rao, P.S. Sastry, A reinforcement learning approach to automatic generation control, Electr. Power Syst. Res. 63 (1) (2002) 9–26.
[109] Tao Yu, B Zhou, Ka Wing Chan, Y Yuan, Bo Yang, QH Wu, R (λ) imitation learning for automatic generation control of interconnected power grids, Automatica 48 (9) (2012) 2130–2136.
[110] Yujian Ye, Dawei Qiu, Mingyang Sun, Dimitrios Papadaskalopoulos, Goran Strbac, Deep reinforcement learning for strategic bidding in electricity markets, IEEE Trans. Smart Grid 11 (2) (2019) 1343–1355.
[111] Sina Zarrabian, Rabie Belkacemi, Adeniyi A. Babalola, Reinforcement learning approach for congestion management and cascading failure prevention with experimental application, Electr. Power Syst. Res. 141 (2016) 179–190.
[112] Xiaorui Liu, Charalambos Konstantinou, Reinforcement learning for cyber-physical security assessment of power systems, in: 2019 IEEE Milan PowerTech, IEEE, 2019, pp. 1–6.
[113] Yin Cheng, Yuexin Huang, Bo Pang, Weidong Zhang, ThermalNet: A deep reinforcement learning-based combustion optimization system for coal-fired boiler, Eng. Appl. Artif. Intell. 74 (2018) 303–311.
[114] Jigao Fu, Hong Xiao, Hao Wang, Junhao Zhou, Control strategy for denitrification efficiency of coal-fired power plant based on deep reinforcement learning, IEEE Access 8 (2020) 65127–65136.
[115] Xianyuan Zhan, Haoran Xu, Yue Zhang, Yusen Huo, Xiangyu Zhu, Honglei Yin, Yu Zheng, Deepthermal: Combustion optimization for thermal power generating units using offline reinforcement learning, 2021, arXiv preprint arXiv:2102.11492.
[116] Volker Stephan, Klaus Debes, H-M Gross, F Wintrich, H Wintrich, A new control scheme for combustion processes using reinforcement learning based on neural networks, Int. J. Comput. Intell. Appl. 1 (02) (2001) 121–136.
[117] OpenAI: Marcin Andrychowicz, Bowen Baker, Maciek Chociej, Rafal Jozefowicz, Bob McGrew, Jakub Pachocki, Arthur Petron, Matthias Plappert, Glenn Powell, Alex Ray, et al., Learning dexterous in-hand manipulation, Int. J. Robot. Res. 39 (1) (2020) 3–20.
[118] Ilge Akkaya, Marcin Andrychowicz, Maciek Chociej, Mateusz Litwin, Bob McGrew, Arthur Petron, Alex Paino, Matthias Plappert, Glenn Powell, Raphael Ribas, et al., Solving rubik's cube with a robot hand, 2019, arXiv preprint arXiv:1910.07113.
[119] Bianca Sangiovanni, Angelo Rendiniello, Gian Paolo Incremona, Antonella Ferrara, Marco Piastra, Deep reinforcement learning for collision avoidance of robotic manipulators, in: 2018 European Control Conference (ECC), IEEE, 2018, pp. 2063–2068.
[120] Steven A Harp, Sergio Brignone, Bruce F Wollenberg, Tariq Samad, SEPIA. a simulator for electric power industry agents, IEEE Control Syst. Mag. 20 (4) (2000) 53–69.
[121] Morteza Rahimiyan, Habib Rajabi Mashhadi, An adaptive Q-learning algorithm developed for agent-based computational modeling of electricity market, IEEE Trans. Syst. Man Cybern. C 40 (5) (2010) 547–556.
[122] Vishnuteja Nanduri, Tapas K. Das, A reinforcement learning model to assess market power under auction-based energy pricing, IEEE Trans. Power Syst. 22 (1) (2007) 85–95.
[123] Byung-Gook Kim, Yu Zhang, Mihaela Van Der Schaar, Jang-Won Lee, Dynamic pricing and energy consumption scheduling with reinforcement learning, IEEE Trans. Smart Grid 7 (5) (2015) 2187–2198.
[124] Thilo Krause, Elena Vdovina Beck, Rachid Cherkaoui, Alain Germond, Goran Andersson, Damien Ernst, A comparison of Nash equilibria analysis and agent-based modelling for power markets, Int. J. Electr. Power Energy Syst. 28 (9) (2006) 599–607.
[125] Sergey Levine, Aviral Kumar, G. Tucker, Justin Fu, Offline reinforcement learning: Tutorial, review, and perspectives on open problems, 2020, ArXiv, abs/2005.01643.
[126] Aviral Kumar, Aurick Zhou, G. Tucker, Sergey Levine, Conservative Q-learning for offline reinforcement learning, in: Advances in Neural Information Processing Systems, 2020.
[127] Xinyue Chen, Zijian Zhou, Z. Wang, Che Wang, Yanqiu Wu, Qing Deng, Keith W. Ross, BAIL: Best-action imitation learning for batch deep reinforcement learning, in: Advances in Neural Information Processing Systems, 2020.
[128] Ilya Kostrikov, Ashvin Nair, Sergey Levine, Offline reinforcement learning with implicit Q-learning, in: International Conference on Learning Representations, 2022.
[129] Lili Chen, Kevin Lu, Aravind Rajeswaran, Kimin Lee, Aditya Grover, Michael Laskin, P. Abbeel, A. Srinivas, Igor Mordatch, Decision transformer: Reinforcement learning via sequence modeling, 2021, ArXiv, abs/2106.01345.
[130] Yue Wu, Shuangfei Zhai, Nitish Srivastava, Joshua M. Susskind, Jian Zhang, Ruslan Salakhutdinov, Hanlin Goh, Uncertainty weighted actor-critic for offline reinforcement learning, in: ICML, 2021.
[131] Zhe Dong, Xiaojin Huang, Yujie Dong, Zuoyi Zhang, Multilayer perception based reinforcement learning supervisory control of energy systems with application to a nuclear steam supply system, Appl. Energy 259 (2020) 114193.
[132] Koroush Shirvan Forgeta, Physics-informed reinforcement learning optimization of nuclear assembly design, 2021.
[133] Xiangyi Chen, Asok Ray, Deep reinforcement learning control of a boiling water reactor, IEEE Trans. Nucl. Sci. 69 (8) (2022) 1820–1832.
[134] Paul Seurin, Koroush Shirvan, Assessment of reinforcement learning algorithms for nuclear power plant fuel optimization, 2023, arXiv preprint arXiv:2305.05812.
[135] Tianhao Zhang, Zhe Dong, Xiaojin Huang, Multi-objective optimization of thermal power and outlet steam temperature for a nuclear steam supply system with deep reinforcement learning. Available at SSRN 4490266.
[136] JaeKwan Park, TaekKyu Kim, SeungHwan Seong, Providing support to operators for monitoring safety functions using reinforcement learning, Prog. Nucl. Energy 118 (2020) 103123.
[137] Daeil Lee, Hyojin Kim, Younhee Choi, Jonghyun Kim, Development of autonomous operation agent for normal and emergency situations in nuclear power plants, in: 2021 5th International Conference on System Reliability and Safety (ICSRS), IEEE, 2021, pp. 240–247.
[138] Jing Li, Yanyang Liu, Xianguo Qing, Kai Xiao, Ying Zhang, Pengcheng Yang, Yue Marco Yang, The application of deep reinforcement learning in coordinated control of nuclear reactors, in: Journal of Physics: Conference Series, Vol. 2113, IOP Publishing, 2021, 012030.
[139] Jae Min Kim, Junyong Bae, Seung Jun Lee, Strategy to coordinate actions through a plant parameter prediction model during startup operation of a nuclear power plant, Nucl. Eng. Technol. 55 (3) (2023) 839–849.
[140] Junyong Bae, Jae Min Kim, Seung Jun Lee, Deep reinforcement learning for a multi-objective operation in a nuclear power plant, Nucl. Eng. Technol. (2023).
[141] Jae Min Kim, Seung Jun Lee, Framework of two-level operation module for autonomous system of nuclear power plants during startup and shutdown operation, in: Transactions of the Korean Nuclear Society Autumn Meeting, 2019.
[142] Daeil Lee, Awwal Mohammed Arigi, Jonghyun Kim, Algorithm for autonomous power-increase operation using deep reinforcement learning and a rule-based system, IEEE Access 8 (2020) 196727–196746.
[143] JaeKwan Park, TaekKyu Kim, SeungHwan Seong, SeoRyong Koo, Control automation in the heat-up mode of a nuclear power plant using reinforcement learning, Prog. Nucl. Energy 145 (2022) 104107.
[144] Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de Las Casas, et al., Magnetic control of tokamak plasmas through deep reinforcement learning, Nature 602 (7897) (2022) 414–419.
[145] K.E. Carlson, P.A. Roth, V.H. Ransom, ATHENA Code Manual. Volume 1. Code Structure, System Models, and Solution Methods, Technical Report, EG and G Idaho, Inc., Idaho Falls (USA), 1986.
[146] Deail Lee, Jonghyun Kim, Autonomous algorithm for start-up operation of nuclear power plants by using LSTM, in: International Conference on Applied Human Factors and Ergonomics, Springer, 2018, pp. 465–475.
[147] Seung Jun Lee, Poong Hyun Seong, Development of automated operating procedure system using fuzzy colored petri nets for nuclear power plants, Ann. Nucl. Energy 31 (8) (2004) 849–869.
[148] Yochan Kima, Jinkyun Park, Envisioning human-automation interactions for responding emergency situations of NPPs: a viewpoint from human-computer interaction, in: Proc. Trans. Korean Nucl. Soc. Autumn Meeting, 2018.
[149] Ar Ryum Kim, Jinkyun Park, Ji Tae Kim, Jaewhan Kim, Poong Hyun Seong, Study on the identification of main drivers affecting the performance of human operators during low power and shutdown operation, Ann. Nucl. Energy 92 (2016) 447–455.
[150] Thomas M Moerland, Joost Broekens, Aske Plaat, Catholijn M Jonker, et al., Model-based reinforcement learning: A survey, Found. Trends® Mach. Learn. 16 (1) (2023) 1–118.
[151] Michael Janner, Justin Fu, Marvin Zhang, Sergey Levine, When to trust your model: Model-based policy optimization, Adv. Neural Inf. Process. Syst. 32 (2019).
[152] Marvin Zhang, Sharad Vikram, Laura Smith, Pieter Abbeel, Matthew Johnson, Sergey Levine, Solar: Deep structured representations for model-based reinforcement learning, in: International Conference on Machine Learning, PMLR, 2019, pp. 7444–7453.

[153] Sergey Levine, Aviral Kumar, George Tucker, Justin Fu, Offline reinforcement learning: Tutorial, review, and perspectives on open problems, 2020, arXiv preprint arXiv:2005.01643.
[154] Yifan Wu, George Tucker, Ofir Nachum, Behavior regularized offline reinforcement learning, 2019, arXiv preprint arXiv:1911.11361.
[155] Rasmus E Andersen, Steffen Madsen, Alexander BK Barlo, Sebastian B Johansen, Morten Nør, Rasmus S Andersen, Simon Bøgh, Self-learning processes in smart factories: Deep reinforcement learning for process control of robot brine injection, Procedia Manuf. 38 (2019) 171–177.
[156] Steven Spielberg, Aditya Tulsyan, Nathan P Lawrence, Philip D Loewen, R Bhushan Gopaluni, Deep reinforcement learning for process control: A primer for beginners, 2020, arXiv preprint arXiv:2004.05490.
[157] Daeil Lee, Seoryong Koo, Inseok Jang, Jonghyun Kim, Comparison of deep reinforcement learning and PID controllers for automatic cold shutdown operation, Energies 15 (8) (2022) 2834.
[158] Jiafei Lyu, Le Wan, Zongqing Lu, Xiu Li, Off-policy RL algorithms can be sample-efficient for continuous control via sample multiple reuse, 2023, arXiv preprint arXiv:2305.18443.
[159] Jiafei Lyu, Xiaoteng Ma, Jiangpeng Yan, Xiu Li, Efficient continuous control with double actors and regularized critics, in: AAAI Conference on Artificial Intelligence, 2021.
[160] Daeil Lee Hee-Jae Lee, Jonghyun Kim, Anomaly recovery algorithm based on robust AI concept for nuclear power plants, 2023.
[161] Daeil Lee, Jonghyun Kim, Concept of robust AI with meta-learning for accident diagnosis, 2022.
[162] Jonah Siekmann, Kevin R. Green, John Warila, Alan Fern, Jonathan W. Hurst, Blind bipedal stair traversal via sim-to-real reinforcement learning, 2021, ArXiv, abs/2105.08328.
[163] Jan Matas, Stephen James, Andrew J. Davison, Sim-to-real reinforcement learning for deformable object manipulation, 2018, ArXiv, abs/1806.07851.
[164] Josiah P. Hanna, Siddharth Desai, Haresh Karnan, Garrett A. Warnell, Peter Stone, Grounded action transformation for sim-to-real reinforcement learning, Mach. Learn. 110 (2021) 2469–2499.
[165] Wenshuai Zhao, Jorge Peña Queralta, Tomi Westerlund, Sim-to-real transfer in deep reinforcement learning for robotics: a survey, in: 2020 IEEE Symposium Series on Computational Intelligence (SSCI), 2020, pp. 737–744.
[166] Han Hu, Kaicheng Zhang, Aaron Hao Tan, Michael Ruan, Christopher Agia, Goldie Nejat, A sim-to-real pipeline for deep reinforcement learning for autonomous robot navigation in cluttered rough terrain, IEEE Robot. Autom. Lett. 6 (2021) 6569–6576.
[167] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, Pieter Abbeel, Sim-to-real transfer of robotic control with dynamics randomization, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2018, pp. 3803–3810.
[168] Stephen James, Paul Wohlhart, Mrinal Kalakrishnan, Dmitry Kalashnikov, Alex Irpan, Julian Ibarz, Sergey Levine, Raia Hadsell, Konstantinos Bousmalis, Sim-to-real via sim-to-sim: Data-efficient robotic grasping via randomized-to-canonical adaptation networks, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 12627–12637.
[169] Karol Arndt, Murtaza Hazara, Ali Ghadirzadeh, Ville Kyrki, Meta reinforcement learning for sim-to-real domain adaptation, in: 2020 IEEE International Conference on Robotics and Automation (ICRA), 2020, pp. 2725–2731.
[170] Jeong-Hoon Lee, Jongeun Choi, Attaining interpretability in reinforcement learning via hierarchical primitive composition, 2021, ArXiv, abs/2110.01833.
[171] Caihua Shan, Yifei Shen, Yao Zhang, Xiang Li, Dongsheng Li, Reinforcement learning enhanced explainer for graph neural networks, in: NeurIPS, 2021.
[172] Francis Maes, Raphaël Fonteneau, Louis Wehenkel, Damien Ernst, Policy search in a space of simple closed-form formulas: Towards interpretability of reinforcement learning, in: Discovery Science, 2012.
[173] Nicklas Hansen, Xiaolong Wang, Generalization in reinforcement learning by soft data augmentation, in: 2021 IEEE International Conference on Robotics and Automation (ICRA), 2021, pp. 13611–13617.
[174] Roberta Raileanu, Rob Fergus, Decoupling value and policy for generalization in reinforcement learning, 2021, ArXiv, abs/2102.10330.
[175] Roberta Raileanu, Maxwell Goldstein, Denis Yarats, Ilya Kostrikov, Rob Fergus, Automatic data augmentation for generalization in reinforcement learning, in: NeurIPS, 2021.
[176] Sam Witty, Jun Ki Lee, Emma Tosch, Akanksha Atrey, Michael L. Littman, David D. Jensen, Measuring and characterizing generalization in deep reinforcement learning, 2021, ArXiv, abs/1812.02868.
[177] Karl Cobbe, Oleg Klimov, Christopher Hesse, Taehoon Kim, John Schulman, Quantifying generalization in reinforcement learning, 2019, ArXiv, abs/1812.02341.