Digital twin based reinforcement learning for extracting network structures and load patterns in planning and operation of distribution systems

Weiqi Hua[a], Bruce Stephen[b], David C.H. Wallom[a]

[a]*Department of Engineering Science, University of Oxford, Oxford OX1 3QG, UK*
[b]*Institute for Energy and Environment, Department of Electronic and Electrical Engineering, University of Strathclyde, Glasgow G1 1RD, UK*

Abstract

Low voltage distribution networks deliver power to the last mile of the network, but are often legacy assets from a time when low carbon technologies, e.g., electrified heat, storage, and electric vehicles, were not envisaged. Furthermore, exploiting emerging data from distribution networks to provide decision support for adapting planning and operational strategies with system transitions presents a challenge. To overcome these challenges, this paper proposes a novel application of digital twins based reinforcement learning to improve decision making by a distribution system operator, with key metrics of predictability, responsiveness, interoperability, and automation. The power system states, i.e., network configurations, technological combinations, and load patterns, are captured via a convolutional neural network, chosen for its pattern recognition capability with high-dimensional inputs. The convolutional neural networks are iteratively trained through the fitted Q-iteration algorithm, as a batch mode reinforcement learning, to adapt the planning and operational decisions with the dynamic system transitions. Case studies demonstrate the effectiveness of the proposed model by reducing 50% of the investment cost when the system transitions towards the winter and maintaining the power loss and loss of load within 5% compared to the benchmark optimisation. Doubled power consumption was observed in winter under future energy scenarios due to the electrification of heat. The trained model can accurately adapt optimal decisions according to the system changes while reducing the computational time of solving optimisation problems, for a range of scales of distribution systems, demonstrating its potential for scalable deployment by a system operator.

Keywords: digital twin, distribution network, fitted Q-iteration, load pattern, network configuration, reinforcement learning.

1. Introduction

Low-voltage distribution networks are the connection between transmission networks and various commercial, light industrial, and residential consumers, i.e., they are the edge of power networks. With the target of net zero energy transition, these consumers increasingly become part of the network, as they actively produce and store energy through the use of distributed energy sources (e.g., solar panels and heat pumps) and storage systems (e.g., batteries and hot water cylinders), which unlocks demand side flexibility and facilitates the distribution networks transitioning towards distribution systems [1]. Nonetheless, a number of challenges arise from the perspective of the distribution system operator (DSO): *(1)* How to exploit data from distribution systems to assist the decision making of the DSO on planning and operation; *(2)* How to adapt decisions on network structures and investments to cope with dynamic transitions of system states, e.g., transition of load patterns; *(3)* How to design a scalable model for planning distribution systems under heterogeneous energy patterns, dynamic network structures, and multiple technological combinations.

To overcome the first challenge, the digitalisation of power systems enables the large volumes of data to be exploited in assisting the strategic planning, accurate prediction, and smart control of distribution systems. Digital twins, a virtual replica of a physical asset, integrate advanced metering, simulation, communication, optimisation, and control technologies [2]. Digital twins are vital for doing all of these things at minimal risks, with the enhanced predictability, responsiveness, interoperability, and automation of distribution systems. For predictability, digital twins are able to accurately predict the state-temporal transitions of distribution systems and take corresponding planning strategies prior to the actual installations, so as to minimise the investment cost and improve the system performance, e.g., reducing power losses. For responsiveness, incorporating digital twins with physical monitoring devices can assist the situational awareness and make active responses to operational signals through taking optimal control decisions, to maintain the stability of distribution systems. For interoperability, digital twins assist the DSO to collectively coordinate stakeholders of distribution systems, in achieving the overall system benefits. For automation, digital twins provide a wide area control and network optimisation to ensure the autonomous operation and self-healing of distribution systems without human interventions.

As indicated by recent research [3–7] and industrial practices [8–10], digital twins are required to accurately represent the operational consequences of planning decisions, based upon physical models and actual data of an observed power system. You

Email addresses: weiqi.hua@eng.ox.ac.uk (Weiqi Hua), bruce.stephen@strath.ac.uk (Bruce Stephen), david.wallom@oerc.ox.ac.uk (David C.H. Wallom)

et al. [3] incorporated machine learning into a predictive digital twin to forecast uncertainties of renewable generation and flexible consumption during the operational scheduling of integrated thermal and electrical energy systems. Saad et al. [4] developed an IoT-based digital twin interacting with the control system to prevent the injection of coordinated false data and the denial of service attacks on interconnected microgrids. Granacher et al. [5] proposed an interoperable digital twin to assist the decision making on the design of energy systems based on the preferences of decision makers. In [6], an energy management tool was designed based on digital twins to provide the services of optimal control, scheduling, forecasting, and coordination for multi-energy systems. Researchers in [7] developed digital twins of cooling, heating, and power systems for both real-time and life-cycle optimisation, in order to improve energy saving and cold recovery. With respect to industrial practices, Powerstar [8] has developed a digital twin to evaluate the feasibility and cost-effectiveness of microgrids, prior to committing investments in the actual installation and implementation. GE Digital [9] has focused on developing the digital twin software based on the operational data of network infrastructures and generators for cost reductions. Siemens [10] has designed an electrical digital twin enabling the power system operators to optimise system performances and automatically exchange data between the internal and external systems. The commons, differences, advantages, and future opportunities of such works on the applications of digital twins in power systems are compared in **Table 1.**

With respect to the second challenge, it is necessary for the DSO to continuously maintain an accurate model, in order to adapt dynamic transitions of system states. The reinforcement learning (RL) fits for this purpose, since a control policy can be learned from interactions with environments. The RL is only driven by historical data without predefined parameters or formulations. Developing approaches of RL for extracting features of power network configurations [11, 12], load patterns [13], and device investments [14] has been well documented in literatures. In [11], a deep RL algorithm was developed for finding optimal topology of distribution networks, in order to increase the renewable integration and reduce the investment cost. Gao et al. [12] proposed a data-driven batch-constrained RL algorithm for the dynamic reconfiguration of distribution networks through learning the control policy with historical operational data. Reference [13] developed a Q-learning based RL for setpoint planning of air conditioning, through which the transitions of load patterns were captured by deep neural networks. In [14], a RL method was proposed based on the Monte Carlo tree search for guiding the DSO to install active/reactive power control devices with reduced planning and operational costs.

For the third challenge, machine learning and deep neural networks are capable of extracting key features from high-dimensional datasets consisting of various types of system states (e.g., network structures, energy patterns, and technological combinations) and temporal transitions (e.g., from the short-term operation to long-term planning), so as to enhance the model scalability with improved computational efficiency. Zhao [15] designed a machine learning selection approach to guide an accurate regression learner-based model for predicting dynamic battery ageing, in order to achieve a trade-off between short-term operational costs and long-term degradation costs. In [16], a data driven machine learning model was proposed to predict building energy demand and enhance flexibility provision with improved computational efficiency. The convolutional neural network (CNN) has the particular potential to generalise the representations of key features from high-dimensional inputs. This is because multiple filers of convolutional layers slide through the input to capture the state correlations and temporal transitions. Claessens et al. [17] implemented the CNN to extract hidden state-time features of residential consumers and developed a RL algorithm to control clusters of thermostatically controlled loads. Kamruzzaman et al. [18] designed a CNN to solve the optimal power flow for various scales of integrated power networks, so as to improve the efficiency of the reliability evaluation. Analogously, the CNN was developed in [19] to evaluate both the transient stability and instability mode of power systems. In [20], a multi-temporal-spatial-scale method was proposed to capture the transitions and characteristics of load patterns through using the CNN.

Although extensive studies have been conducted to address those three challenges, there are three major gaps as: *(1)* Existing research has focused on the application of digital twins to exploit data from physical power systems in assisting decision making, but incorporating both planning (e.g., technology investment or network configuration prior to setting-up a new distribution system) and operation (e.g., maintaining system constraints in parallel with the real operation) of a distribution system is missing. *(2)* For the research using RL to adapt transitions of a distribution system, there is still a lack of a model which includes the network structures, load patterns, and technological combinations together. *(3)* Although the research efforts have been dedicated to improving the model scalability and computational efficiency, how to reflect the states and characteristics of a distribution system for the CNN to extract feature representations has not been well explored.

By filling the identified gaps in the existing research, this paper offers the following key contributions:

- A framework of digital twins is designed in combination with the RL to adapt decisions of the DSO with dynamic transitions of system states, through which the RL continuously improves the model accuracy through exploiting the data provided by digital twins.
- A CNN is tailored to extract feature representations from the input information of network configurations, technology installations, and load patterns under various scales of distribution networks, and map these feature representations to the optimal policy of planning and operation for distribution systems.
- The proposed model was implemented in the IEEE 33-bus, 18-bus, and 69-bus distribution networks to demonstrate the scalability. The designed RL algorithm can accurately make decisions which are close to those from solving the

Table 1 Comparison of research and industrial practices on the application of digital twins in power systems.

	You et al. [3]	Saad et al. [4]	Granacher et al. [5]	Dwyer et al. [6]	Huang et al. [7]	Powerstar [8]	GE Digital [9]	Siemens [10]
Function	Prediction	Protection	Design	Scheduling and forecasting	Scheduling	Investment	Scheduling	System performance
Mode	Operation	Operation	Planning	Operation	Operation	Planning	Operation	Operation
Context	Integrated thermal and electrical energy system	Interconnected microgrids	Integrated biorefinery	Integrated energy systems	Cooling, heating, and power system	Microgrids	Generators and infrastructures	Power system
Advantage	Cost saving	Preventing cyber attacks	Considering users' preferences	Real-time parallel evaluation	Energy saving and cold recovery	Prior to actual installation	Cost reduction	System optimisation and automation
Common	Application of digital twins to exploit data from physical power systems in assisting decision making							
Opportunity	Developing digital twins incorporating planning and operation of power distribution systems, both prior to setting-up a new system and in parallel with real operation							

problems of minimising the investment cost, power loss, loss of load, and renewable curtailment. A better adaption with the system transition was also found compared to solving the minimisation problems in each single stage.

The rest of this paper is organised as follows: Section 2 introduces the overview framework for implementing the designed digital twin based RL model. A Markov decision process is detailed in Section 3 to discuss the states, actions, and costs when the DSO makes sequential planning and operational decisions. The algorithms of the RL are provided in Section 4. In Section 5, the performances on learning, model outputs, and scalability are evaluated by case studies. The research limitations, challenges and prospects are discussed in Section 6. Section 7 concludes this paper and lists potential future works.

2. Framework

This section introduces the overall framework for implementing the designed digital twin based RL model. The objective of this model is to adapt the planning and operational decisions of the DSO with dynamic transitions of network configurations, technology installations, and load patterns. This objective is achieved by exploiting the data from physical distribution systems through using the RL to continuously improve the model accuracy and scalability.

The overall framework for implementing the designed model is presented in **Fig. 1**. *First*, the information of system states, including the network configurations, technology installations, and load patterns from the physical distribution systems is fed into the digital twin. *Second*, under these states, the DSO can test a set of planning actions and solve the optimal operational control problems with the objectives of minimising the investment cost, power loss, renewable curtailment, and loss of load to yield the control decisions. The values of four objective functions are taken as the cost to judge how good those tested actions are. This process iteratively proceeds until finding the best planning and operational decisions. *Third*, the fitted Q-iteration (FQI), as a batch mode RL algorithm, is used to yield optimal Q-function for the DSO through iteratively training neural networks using the states and actions as inputs. The optimal Q-function is used to guide the DSO to perform the planning and operational control decisions in physical distribution systems.

The designed digital twins provide the DSO with the following five primary functions: *(1)* Collecting information of system states from sensors and smart meters of a physical distribution system; *(2)* Prior to setting-up a new distribution system, testing any potential planning decisions for the DSO, in achieving certain objectives; *(3)* In parallel with real operations, running the power flow analysis and examining violations of system constraints; *(4)* Systematically integrating data analytics and RL algorithms for the network optimisation of DSO; *(5)* Automatically sending signals to control network configurations (through circuit barkers and switchers) and operations (through connecting/disconnecting loads, renewable energy sources, capacitor banks, static VAR compensators (SVCs), and storages).

3. Markov decision process

The decision making of the DSO on the planning and operation of a distribution system is modelled as a Markov decision process. The Markov decision process is determined by its action, state, and state transition function as

$$\mathbf{S}_{k+1} = f^{\mathrm{tr}}\left(\mathbf{S}_k, \mathbf{A}_k\right), \forall k \in \mathcal{K}, \tag{1}$$

where $\mathbf{S}_k$ is the state of the Markov decision process at the stage k, $\mathbf{A}_k$ is the action of the Markov decision process at the stage k, $f^{\mathrm{tr}}(\cdot)$ is the state transition function describing the dynamics from the state $\mathbf{S}_k$ to $\mathbf{S}_{k+1}$, and $\mathcal{K}$ is the space of all stages.

Taking an action $\mathbf{A}_k$ under the state $\mathbf{S}_k$ would result in a cost, denoted as $c\left(\mathbf{S}_k, \mathbf{A}_k\right)$. The objective of the DSO is to find a control policy which minimises the cumulative future costs over $|\mathcal{K}|$ stages as

$$u_k := \sum_{k=1}^{|\mathcal{K}|} \left[\gamma^{(k-1)} \cdot c\left(\mathbf{S}_k, \mathbf{A}_k\right)\right], \tag{2}$$

where u_k is the cumulative future cost, γ is the discount factor which reflects the importance of the cost in the distant future relative to that in the immediate future.

Until the stage k, the only observations of states and actions are $(\mathbf{S}_1, ..., \mathbf{S}_k)$ and $(\mathbf{A}_1, ..., \mathbf{A}_k)$, respectively, whereas the future states $(\mathbf{S}_{k+1}, ..., \mathbf{S}_{|\mathcal{K}|})$ and actions $(\mathbf{A}_{k+1}, ..., \mathbf{A}_{|\mathcal{K}|})$ are uncertain. The action value function, i.e., Q-function, can estimate these uncertain variables by calculating the expectation of the cumulative future costs u_k with respect to $(\mathbf{S}_{k+1}, ..., \mathbf{S}_{|\mathcal{K}|})$ and $(\mathbf{A}_{k+1}, ..., \mathbf{A}_{|\mathcal{K}|})$ as

$$q_\pi\left(\mathbf{S}_k, \mathbf{A}_k\right) := \mathbb{E}\left(u_k | \mathbf{S}_k, \mathbf{A}_k\right), \tag{3}$$

where $q_\pi(\cdot)$ is the Q-function which quantifies the performance of taking the action $\mathbf{A}_k$ under the state $\mathbf{S}_k$, following a given policy function $\pi(\cdot)$. The policy function maps a given state to the corresponding action as

$$\mathbf{A}_k = \pi\left(\mathbf{S}_k\right), \forall k \in \mathcal{K}. \tag{4}$$

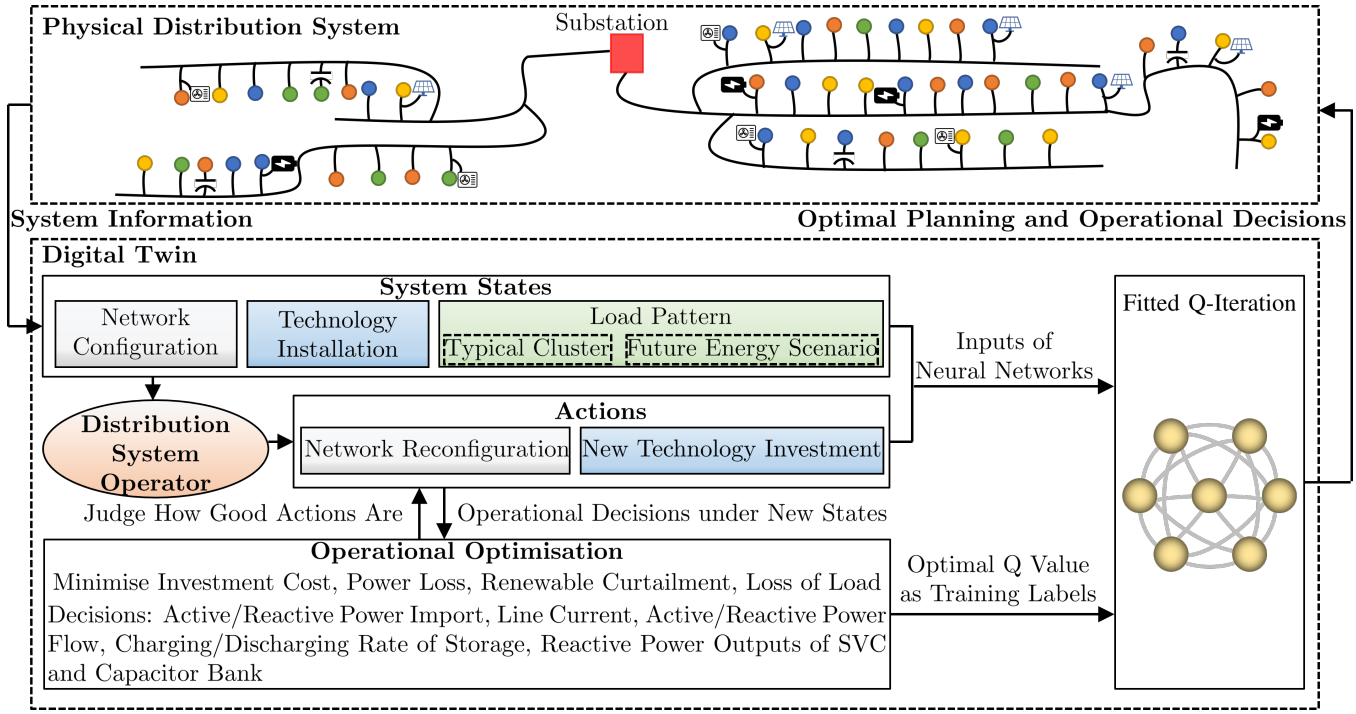

Fig. 1. Framework for implementing the designed digital twin based reinforcement learning model. The states of physical distribution systems are fed into the digital twin, under which the distribution system operator can test planning actions and solve optimal operational control problems to yield a cost. The states, actions, and optimal Q values are used to iteratively train neural networks through the fitted Q-iteration algorithm. The final optimal decisions are performed by the physical distribution systems.

The optimal Q-function is the minimum Q-function obtained by any policy as $q^*(\mathbf{S}_k, \mathbf{A}_k) = \min_\pi q_\pi(\mathbf{S}_k, \mathbf{A}_k)$, which can be obtained by the Bellman optimality equation [21] as

$$q^*(\mathbf{S}_k, \mathbf{A}_k) = \mathbb{E}\left[c(\mathbf{S}_k, \mathbf{A}_k) + \min_{\mathbf{A}_{k+1}} q^*(\mathbf{S}_{k+1}, \mathbf{A}_{k+1})\right]. \quad (5)$$

Therefore, the optimal action can be yielded by

$$\mathbf{A}_k^* = \pi^*(\mathbf{S}_k) = \arg\min_{\mathbf{A}_k} q^*(\mathbf{S}_k, \mathbf{A}_k). \quad (6)$$

3.1. State description

The state space consists of the information on *(1)* network configuration, *(2)* technology installation, and *(3)* load pattern. The advantages of considering these three elements together are that they can more accurately reflect key characteristics of a physical distribution system and cover essential decisions of the DSO. However, the challenge is how to mathematically describe these elements and use them to reinforce the decision making of the DSO. Our designed approach processes these three elements in the form of matrices, which is detailed in the following subsections.

3.1.1. Network configuration

First, how to use a matrix to describe the state of network configurations is introduced. Let $\mathcal{F}$ denote the set of feeders in a distribution network, the state of the network configuration at the stage k can be described by a $(|\mathcal{N}|\times|\mathcal{F}|)$-dimensional matrix $\mathbf{S}_k^{\mathrm{f}}$, in which the element $s_{n,f} \in \mathbf{S}_k^{\mathrm{f}}$ is a binary value indicating whether the bus n is connected to the feeder f (if $s_{n,f}$=1), or not (if $s_{n,f}$=0). **Fig. 2** presents an example of how the matrix $\mathbf{S}_k^{\mathrm{f}}$ represents the configurations of a distribution network with the state transitioning form the stage $k-1$ to the stage k.

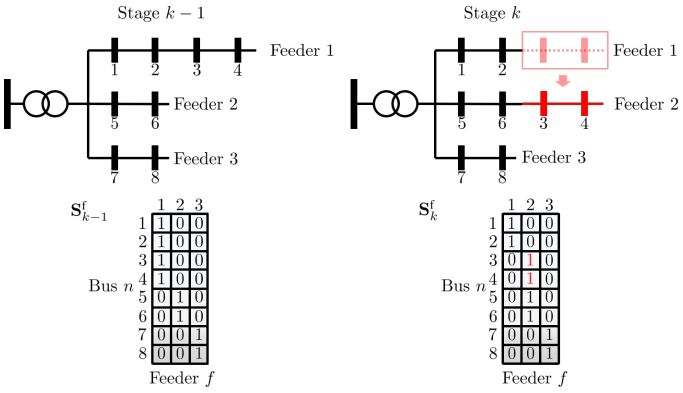

Fig. 2. Schematic illustration of the network configurations transitioning from the stage $k-1$ to the stage k, represented by the matrices $\mathbf{S}_{k-1}^{\mathrm{f}}$ and $\mathbf{S}_k^{\mathrm{f}}$, respectively. Element 1 of the matrices indicates that the bus n (indicated by the vertical dimension) is connected to the feeder f (indicated by the horizontal dimension), and element 0 of the matrices indicates that the bus n is not connected to the feeder f.

Next, once the network configuration at one stage is determined, the resistance and reactance between neighbouring buses need to be measured to calculate the incurred power losses. Since the accurate geographical locations of buses are unknown, our research assumes that the distance and resistance between buses are uniformly distributed along the feeder

length. Hence, the length of the branch from the head of a feeder to the bus n is described as

$$\zeta_n = \frac{\zeta_f}{|\mathcal{N}_f|} \cdot n, \forall n \in \mathcal{N}_f, \tag{7}$$

where ζ_n is the length of the branch from the head of a feeder to the bus n, ζ_f is the total length of the feeder f, and $\mathcal{N}_f$ is the set of buses connected to the feeder f. The resistance between two neighbouring buses $n-1$ and n can be described as

$$r_{n-1,n} = (\zeta_{n-1} - \zeta_n) \cdot r^{\mathrm{u}}, \tag{8}$$

where $r_{n-1,n}$ is the resistance of the branch from the bus $n-1$ to the bus n, and r^{u} is the unit resistance of a feeder. Analogously, the reactance between two neighbouring buses $n-1$ and n can be described as

$$x_{n-1,n} = (\zeta_{n-1} - \zeta_n) \cdot x^{\mathrm{u}}, \tag{9}$$

where $x_{n-1,n}$ is the reactance of the branch from the bus $n-1$ to the bus n, and x^{u} is the unit reactance of a feeder.

3.1.2. Technology installation

In this research, the technologies to be invested by the DSO for installations in a distribution network include the SVC, capacitor bank, and energy storage systems. The SVC and capacitor bank provide fast-acting reactive power to enhance the hosting capacity of distributed generation for ensuring the power quality and stability of distribution systems. The energy storage systems provide the flexibility for the generation, consumption, and import from transmission networks through strategically charging and discharging the active power.

Let $\mathcal{E}$ denote the set of these technologies, the state of the technology installation at the stage k can be described by a $(|\mathcal{N}|\times|\mathcal{E}|)$-dimensional matrix $\mathbf{S}^{\mathrm{e}}_k$, in which the element $s_{n,e} \in \mathbf{S}^{\mathrm{e}}_k$ is an integer indicating the number of the technology e connected to the bus n. **Fig. 3** presents an example of how the matrix $\mathbf{S}^{\mathrm{e}}_k$ represents the technology installations of a distribution network with the state transitioning form the stage $k-1$ to the stage k.

3.1.3. Load pattern

The state of load patterns reflects the composition of heterogeneous consumers, e.g., businesses, households, and industries. *First*, the basic patterns of electricity consumption are captured and clustered as the typical load patterns. *Next*, according to the future energy scenarios of the National Grid ESO [22], there will be increasing flexibility provision from the demand side through the uptake of the roof-top solar panels, electrified heat, electric vehicles, and smart control systems. These components will be modelled and integrated into the identified typical load patterns when distribution systems transition towards future states.

• Cluster of load profile

To capture the recurring load profile patterns and more specifically, the diversity of peak loads, this research develops a statistical approach, i.e., the cluster-producing-merging (CPM),

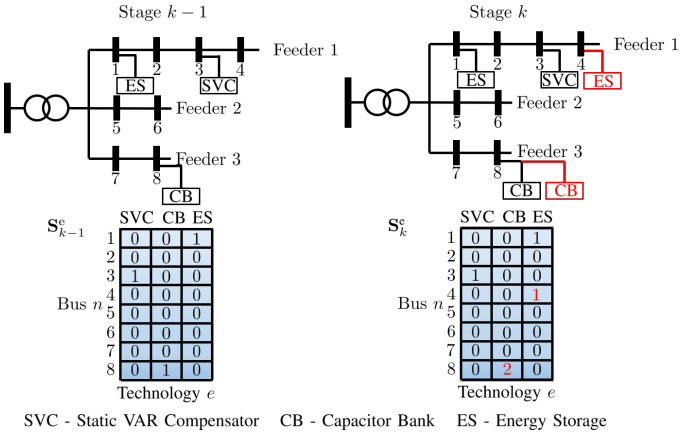

Fig. 3. Schematic illustration of the technology installations transitioning from the stage $k-1$ to the stage k, represented by the matrices $\mathbf{S}^{\mathrm{e}}_{k-1}$ and $\mathbf{S}^{\mathrm{e}}_k$, respectively. Elements of the matrices indicate the total number of the technology e (indicated by the horizontal dimension) connected to the bus n (indicated by the vertical dimension).

to identify typical load patterns of consumers from the historical data of power consumption. The procedures of the CPM are detailed as follows:

Step 1 (Data collection and normalisation): The real-time metering data of power consumption from heterogeneous consumers, is collected from digital twins as historical samples and expressed in the per unit (p.u.) of the active power, considering that certain load profiles would have the similar shape but different magnitudes.

Step 2 (Probability estimation): The non-parametric kernel density estimation [23] is used to fit the probability density function (pdf) from the historical samples of power consumption at each time step. As a non-parametric approach, the kernel density estimation can accurately reflect the distributions of historical samples without the need of predefined parameters. The fitted pdf can be described as

$$\hat{f}^{\mathrm{pdf}}\left(p^{\mathrm{load}}_t\right) := \frac{1}{|\mathcal{H}| \cdot \kappa} \cdot \sum_{h \in \mathcal{H}} f^{\mathrm{k}}\left(\frac{p^{\mathrm{load}}_t - p^{\mathrm{load}}_{t,h}}{\kappa}\right), \tag{10}$$

where $\hat{f}^{\mathrm{pdf}}(\cdot)$ is the fitted pdf, p^{load}_t is the uncertain variable of power consumption at the time step t, $p^{\mathrm{load}}_{t,h}$ is the historical sample h of the uncertain variable p^{load}_t, $\mathcal{H}$ is the set of historical samples of the power consumption, κ is the bandwidth smoothing parameter, and $f^{\mathrm{k}}(\cdot)$ is the kernel function. Each kernel function is placed over a sample. Hence, the pdf is fitted by the sum of $|\mathcal{H}|$ kernels.

Step 3 (Cluster producing): The Latin hypercube sampling [24] is used to produce initial clusters with two key functions: *(1)* When the size of historical data is limited, it can be used for data augmentation to avoid overfitting; *(2)* It can capture the potential variations of uncertain power consumption based on the generated density function. Let $\mathcal{Y}$ denote the set of clusters. For the initially produced clusters, we have $|\mathcal{Y}|>|\mathcal{H}|$. To produce $|\mathcal{Y}|$ clusters, the cumulative distribution function (cdf) derived from the pdf is equally divided into $|\mathcal{Y}|$ intervals. Each cluster is randomly sampled from each of these intervals and

then calculated as the inverse function of the cdf as

$$p_{t,y}^{\text{load}} = \left[f^{\text{cdf}}\left(p_t^{\text{load}}\right)|_{p_t^{\text{load}}=p_{t,y}^{\text{load}}}\right]^{-1} = \left[\left(\frac{1}{|\mathcal{Y}|}\right)\cdot\sigma + \frac{y-1}{|\mathcal{Y}|}\right]^{-1}, \quad (11)$$

where $p_{t,y}^{\text{load}}$ is the cluster y of power consumption at the time step t, $f^{\text{cdf}}(\cdot)$ is the cdf, and $\sigma \in [0,1]$ is a random variable being subject to the uniform distribution. For the set of initially produced clusters, the occurrence probability of every cluster is the same as $\mathbb{P}\left(p_{t,y}^{\text{load}}\right) = 1/|\mathcal{Y}|$.

Step 4 (Cluster merging): To identify the high-probable clusters, i.e., typical energy patterns, the similar clusters will be merged and low-probable clusters will be dropped. The similarity between any two clusters y_1 and y_2 are measured by the Euclidean distance [25] as

$$l(y_1, y_2) := \sqrt{\left(p_{t,y_1}^{\text{load}} - p_{t,y_2}^{\text{load}}\right)^2}, \quad (12)$$

where $l(\cdot)$ is the function of the Euclidean distance between any two clusters. Each cluster y has $|\mathcal{Y}|-1$ Euclidean distance with other clusters y' ($\forall y' \in \mathcal{Y}, y' \neq y$). The minimum Euclidean distance for each cluster is denoted as

$$l(y, y')^* = \min_{y'} l(y, y'). \quad (13)$$

If a cluster has a high similarity, i.e., the minimum Euclidean distance, and low occurrence probability, this cluster will be merged with its similar cluster by adding the occurrence probabilities of these two clusters. Such a cluster is located by finding the minimum product between the occurrence probability and minimum Euclidean distance as

$$y^* = \arg\min_{y}\left[\mathbb{P}\left(p_{t,y}^{\text{load}}\right)\cdot l(y, y')^*\right]. \quad (14)$$

The probability of the cluster y^* is added to the probability of the cluster y' and then the cluster y^* will be removed from the set $\mathcal{Y}$. This removing process iteratively proceeds until reaching the desired number of clusters.

Step 5 (State representation): The state of the load pattern at the stage k can be described by a $(|\mathcal{N}| \times |\mathcal{Y}|)$-dimensional matrix $\mathbf{S}_k^{\text{y}}$, in which the element $s_{n,y} \in \mathbf{S}_k^{\text{y}}$ is a binary value indicating whether the power profile of the bus n matches to the load pattern of the cluster y (if $s_{n,y}$=1), or not (if $s_{n,y}$=0). Every bus in a distribution network will be assigned by one of clusters of power profiles. **Fig. 4** presents the relationship between the clusters of load profiles $p_{t,y}^{\text{load}}, \forall t \in \mathcal{T}, y \in \mathcal{Y}$ and state matrix of the load pattern $\mathbf{S}_k^{\text{y}}$. As an example indicated in the figure, the bus 4 matches to the cluster 2, i.e., $s_{4,2} = 1$ (dashed red box on the right hand side), which means the daily power profile of the bus 4 follows the cluster 2 (dashed red box on the left hand side).

• Future energy scenarios

To investigate the impacts of the demand side flexibility on the planning and operation of distribution systems. The roof-top solar panels, air-source heat pumps, and electric vehicles are integrated into certain clusters of load patterns as

$$p_{t,y}^{\text{load,net}} = p_{t,y}^{\text{load}} - \varsigma_y^{\text{pv}}\cdot p_t^{\text{pv}} + \varsigma_y^{\text{ev}}\cdot p_t^{\text{ev}} + \varsigma_y^{\text{hp}}\cdot p_t^{\text{hp}}, \quad (15)$$

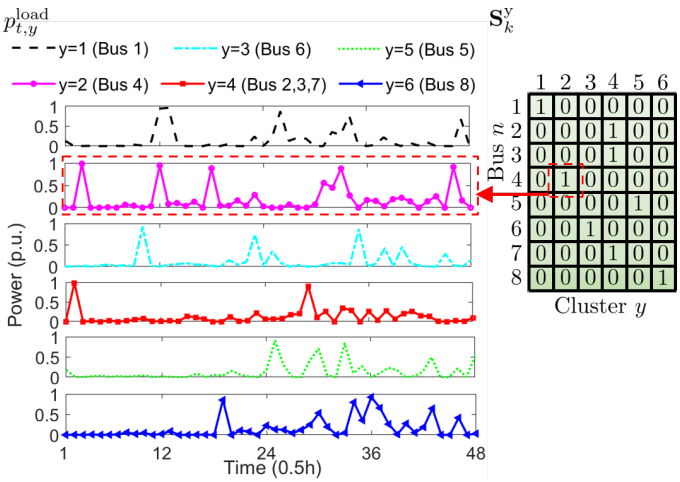

Fig. 4. Schematic illustration of the relationship between the clusters of load profiles $p_{t,y}^{\text{load}}, \forall t \in \mathcal{T}, y \in \mathcal{Y}$ and state matrix of the load pattern $\mathbf{S}_k^{\text{y}}$, and how the matrix $\mathbf{S}_k^{\text{y}}$ represents the load patterns of a distribution network. Element 1 of the matrix indicates that the power profile of the bus n (indicated by the vertical dimension) matches to the load pattern of the cluster y (indicated by the horizontal dimension), and element 0 of the matrix indicates that the power profile of the bus n does not match to the load pattern of the cluster y.

where $p_{t,y}^{\text{load,net}}$ is the net demand at the time step t of the cluster y with the integration of the roof-top solar panels, air-source heat pumps, and electric vehicles, p_t^{pv} is the power output of the roof-top solar panel at the time step t, p_t^{ev} is the power charging to the electric vehicle at the time step t, p_t^{hp} is the electricity used to run the air-source heat pump at the time step t, and ς_y^{pv}, ς_y^{ev}, and ς_y^{hp} are binary values indicating if the roof-top solar panel, air-source heat pump, or electric vehicle is installed (if ς_y^{pv}=1, ς_y^{ev}=1, or ς_y^{hp}=1), or not (if ς_y^{pv}=0, ς_y^{ev}=0, or ς_y^{hp}=0), respectively.

In the future energy scenarios, a consumer would be equipped with a smart control system to strategically switch on/off the air-source heat pump, in order to maintain the indoor temperature within the comfort range. The thermal inertia of a consumer's promises is modelled by the 1R1C thermal model [26] as

$$\begin{aligned}\tau_t =& \exp\left(\frac{-\Delta t\cdot\mu}{\nu}\right)\cdot\tau_{t-1} + \left[1-\exp\left(\frac{-\Delta t\cdot\mu}{\nu}\right)\right]\cdot\tau_t^{\text{a}} \\ &+ \frac{1}{\mu}\cdot\left[1-\exp\left(\frac{-\Delta t\cdot\mu}{\nu}\right)\right]\cdot p_t^{\text{hp}}\cdot\theta\cdot\varepsilon,\end{aligned} \quad (16)$$

where τ_t is the indoor temperature at the time step t, μ is the thermal transmittance of a consumer's premise, ν is the thermal capacitance of a consumer's promise, τ_t^{a} is the temperature of the ambient air at the time step t, θ is the coefficient of performance of the air-source heat pump, and ε is a binary variable indicating whether the air-source heat pump is switched on (if ε=1) or off (if ε=0). To maintain the comfort of a consumer and save energy consumption, the control function is automatically performed according to

$$\varepsilon = \begin{cases}1, & \text{if } \tau_t \leq \tau^{\min},\\ 0, & \text{if } \tau_t > \tau^{\min},\end{cases} \quad (17)$$

where $\tau^{\min}$ is the minimum boundary of the indoor temperature to maintain a consumer's comfort.

3.2. Action description

The action space consists of the information on *(1)* network reconfiguration and *(2)* new technology investment.

First, the action of the network reconfiguration is described by a matrix $\mathbf{A}_k^{\mathrm{f}}$ in the same shape as the state matrix of the network configuration $\mathbf{S}_k^{\mathrm{f}}$, in which the element $a_{n,f} \in \mathbf{A}_k^{\mathrm{f}}$ is a binary value indicating whether the bus n is connected to the feeder f (if $a_{n,f}$=1), or not (if $a_{n,f}$=0). When an action of the network reconfiguration is taken, the state of the network configuration transitions to the next state as

$$\mathbf{S}_{k+1}^{\mathrm{f}} = \mathbf{A}_k^{\mathrm{f}}. \tag{18}$$

Considering that each bus can only connect to and at least connect to one feeder, the sum of elements in each row of the matrix $\mathbf{A}_k^{\mathrm{f}}$ should equal to 1 as

$$\sum_{n} a_{n,f} = 1, \forall f \in \mathcal{F}. \tag{19}$$

Second, the action of the new technology investment is described by a matrix $\mathbf{A}_k^{\mathrm{e}}$ in the same shape as the state matrix of the technology installation $\mathbf{S}_k^{\mathrm{e}}$, in which the element $a_{n,e} \in \mathbf{A}_k^{\mathrm{e}}$ is a binary value indicating whether the technology e is connected to the bus n (if $a_{n,e}$=1), or not (if $a_{n,e}$=0). When an action of the new technology investment is taken, the state of the technology installation transitions to the next state as

$$\mathbf{S}_{k+1}^{\mathrm{e}} = \mathbf{A}_k^{\mathrm{e}} + \mathbf{S}_k^{\mathrm{e}}. \tag{20}$$

It is noted that the state of load patterns are exogenous information which is not influenced by the DSO's actions.

3.3. Cost function

This subsection describes how the cost of RL drives the adaptation of planning and operational control decisions with transitions of a distribution system.

3.3.1. Operation

In the operational phase, the DSO aims to minimise the investment cost, power loss, renewable curtailment, and load curtailment by determining the active/reactive power import, bus voltage, active/reactive power flow, charging/discharging rate of storage, and reactive power outputs of the SVC and capacitor bank, which leads to a multi-objective optimisation problem. The objective functions are detailed as follows.

• The investment cost can be defined as

$$c^{\mathrm{inv}} := \sum_{n \in \mathcal{N}} \mathbf{A}_k^{\mathrm{e}} \cdot \mathbf{c}^{\mathrm{inv}}, \tag{21}$$

where c^{inv} is the investment cost, $\mathbf{c}^{\mathrm{inv}}$ is the $|\mathcal{E}|$-dimensional column vector with the element $c_e \in \mathbf{c}^{\mathrm{inv}}$ to denote the cost coefficient of the technology e, and $\mathcal{N}$ is the index set of buses of a distribution system.

• The total power loss of a distribution network can be defined as

$$p^{\mathrm{loss}} := \sum_{n \in \mathcal{N}, t \in \mathcal{T}} \tilde{i}_{m,n,t} \cdot r_{m,n}, \tag{22}$$

where p^{loss} is the total power loss of a distribution network, $\tilde{i}_{m,n,t}$ is the square of the current magnitude over the branch from the bus m to the bus n at the time step t, $r_{m,n}$ is the resistance of the branch from the bus m to the bus n, and $\mathcal{T}$ is the index set of time steps.

• The total amount of the solar power curtailment of a distribution network can be defined as

$$\Delta p^{\mathrm{pv}} := \sum_{n \in \mathcal{N}, t \in \mathcal{T}} \Delta p_{n,t}^{\mathrm{pv}}, \tag{23}$$

where Δp^{pv} is the total amount of the solar power curtailment of a distribution network, $\Delta p_{n,t}^{\mathrm{pv}}$ is the amount of the solar power curtailment in the bus n at the time step t.

• The total amount of the load curtailment of a distribution network can be defined as

$$\Delta p^{\mathrm{load}} := \sum_{n \in \mathcal{N}, t \in \mathcal{T}} \Delta p_{n,t}^{\mathrm{load}}, \tag{24}$$

where Δp^{load} is the total amount of the load curtailment of a distribution network, and $\Delta p_{n,t}^{\mathrm{load}}$ is the amount of the load curtailment in the bus n at the time step t.

The formulations of objective functions in our research are analogous to the work in [27], but the difference is that authors in [27] took the power loss, renewable curtailment, and load curtailment as constraints.

When a DSO solves the multi-objective optimisation problem, the following constraints [28] need to be considered:

• Power flow constraints: Eq. (25) describes the constraint of the active power flow, and Eq. (26) describes the constraint of the reactive power flow.

$$p_{n,t}^{\mathrm{im}} + p_{n,t}^{\mathrm{es}} - p_{n,t}^{\mathrm{load,net}} - \Delta p_{n,t}^{\mathrm{pv}} + \Delta p_{n,t}^{\mathrm{load}} = \sum_{o \in \delta(n^-)} p_{n,o,t} - \sum_{m \in \delta(n^+)} \left(p_{m,n,t} - \tilde{i}_{m,n,t} \cdot r_{m,n} \right), \forall n \in \mathcal{N}, t \in \mathcal{T}, \tag{25}$$

$$q_{n,t}^{\mathrm{im}} + q_{n,t}^{\mathrm{svc}} + q_{n,t}^{\mathrm{cb}} - \left(q_{n,t}^{\mathrm{load}} - \eta \cdot \Delta p_{n,t}^{\mathrm{load}} \right) = \sum_{o \in \delta(n^-)} q_{n,o,t} - \sum_{m \in \delta(n^+)} \left(q_{m,n,t} - \tilde{i}_{m,n,t} \cdot x_{m,n} \right), \forall n \in \mathcal{N}, t \in \mathcal{T}, \tag{26}$$

where $p_{n,t}^{\mathrm{im}}$ and $q_{n,t}^{\mathrm{im}}$ are the active and reactive power imported from transmission networks through transformers to the bus n at the time step t, respectively, $p_{n,t}^{\mathrm{es}}$ is the active power discharged (when positive)/charged (when negative) from/to the energy storage system in the bus n at the time step t, $p_{n,t}^{\mathrm{load,net}}$ is the net active power demand of loads connected to the bus n at the time step t, $q_{n,t}^{\mathrm{load}}$ is the reactive power demand of loads connected to the bus n at the time step t, $q_{n,t}^{\mathrm{svc}}$ is the reactive power output of the SVC in the bus n at the time step t, $q_{n,t}^{\mathrm{cb}}$ is the reactive power output of the capacitor bank in the bus n at the time

step t, η is the ratio of reactive power to active power of a load. $\delta(n^-)$ is the index set of all the outflowing buses from the bus n, $\delta(n^+)$ is the index set of all the inflowing buses to the bus n, $p_{n,o,t}$ and $q_{n,o,t}$ are the active and reactive power flows from the bus n to the bus o at the time step t, $p_{m,n,t}$ and $q_{m,n,t}$ are the active and reactive power flows from the bus m to the bus n at the time step t, and $x_{m,n}$ is the reactance of the branch from the bus m to the bus n. It is noted that $p^{\text{im}}_{n,t}$ and $q^{\text{im}}_{n,t}$ are positive only if the bus n is connected to a transformer; Otherwise, $p^{\text{im}}_{n,t}$=$q^{\text{im}}_{n,t}$=0.

• Bus voltage constraints: Eq. (27) is the voltage limit of a bus, and Eq. (28) is the voltage difference between two buses.

$$\left(v_n^{\min}\right)^2 \le \tilde{v}_{n,t} \le \left(v_n^{\max}\right)^2, \forall n \in \mathcal{N}, t \in \mathcal{T}, \tag{27}$$

$$\begin{aligned}\tilde{v}_{m,t} - \tilde{v}_{n,t} =& 2 \cdot \left(p_{m,n,t} \cdot r_{m,n} + q_{m,n,t} \cdot x_{m,n}\right) - \\ & \tilde{i}_{m,n,t} \cdot \left[(r_{m,n})^2 + (x_{m,n})^2\right], \forall n, m \in \mathcal{N}, t \in \mathcal{T},\end{aligned} \tag{28}$$

where $v_n^{\min}$ and $v_n^{\max}$ are the minimum and maximum voltage limits of the bus n, and $\tilde{v}_{n,t}$ is the square of the voltage magnitude in the bus n at the time step t. We have $\tilde{v}_{n,t} = (v_{n,t})^2$.

• Line current constraint:

$$0 \le \tilde{i}_{m,n,t} \le \left(i_{m,n}^{\max}\right)^2, \tag{29}$$

where $i_{m,n}^{\max}$ is the maximum line current limit over the branch from the bus m to the bus n. We have $\tilde{i}_{m,n,t} = \left[(p_{m,n,t})^2 + (q_{m,n,t})^2\right] / \tilde{v}_{n,t}$.

• Power constraints: Eq. (30) is the constraint of the generation curtailment, Eq. (31) is the constraint of the load curtailment, Eq. (32) is the active power constraint of a transformer, and Eq. (33) is the reactive power constraint of a transformer.

$$0 \le \Delta p_{n,t}^{\text{pv}} \le p_{n,t}^{\text{pv}}, \forall n \in \mathcal{N}, t \in \mathcal{T}, \tag{30}$$

$$0 \le \Delta p_{n,t}^{\text{load}} \le p_{n,t}^{\text{load,net}}, \forall n \in \mathcal{N}, t \in \mathcal{T}, \tag{31}$$

$$p^{\text{tr,min}} \le \sum_{n \in \delta(\text{tr})} p_{n,t}^{\text{im}} \le p^{\text{tr,max}}, \forall n \in \mathcal{N}, t \in \mathcal{T}, \tag{32}$$

$$q^{\text{tr,min}} \le \sum_{n \in \delta(\text{tr})} q_{n,t}^{\text{im}} \le q^{\text{tr,max}}, \forall n \in \mathcal{N}, t \in \mathcal{T}, \tag{33}$$

where $p_{n,t}^{\text{pv}}$ is the power output of the solar panel connected to the bus n at the time step t, $p^{\text{tr,min}}$ and $p^{\text{tr,max}}$ are the minimum and maximum active power limits of a transformer, respectively, $q^{\text{tr,min}}$ and $q^{\text{tr,max}}$ are the minimum and maximum reactive power limits of a transformer, respectively, and $\delta(\text{tr})$ is the index set of the buses connected to a transformer.

• Energy storage constraints: Eq. (34) is the constraint of the charging/discharging rate of an energy storage system, Eq. (35) is the dynamics of an energy storage system, and Eq. (36) is the capacity constraint of an energy storage system.

$$0 \le \left|p_{n,t}^{\text{es}}\right| \le p_n^{\text{es,max}}, \forall n \in \mathcal{N}, t \in \mathcal{T}, \tag{34}$$

$$e_{n,t}^{\text{es}} = e_{n,t-1}^{\text{es}} - \vartheta \cdot p_{n,t}^{\text{es}} \cdot \Delta t, \forall n \in \mathcal{N}, t \in \mathcal{T}, \tag{35}$$

$$0 \le e_{n,t}^{\text{es}} \le e_n^{\text{es,max}}, \forall n \in \mathcal{N}, t \in \mathcal{T}, \tag{36}$$

where $p_n^{\text{es,max}}$ is the maximum charging/discharging rate of the energy storage system connected to the bus n, $e_{n,t}^{\text{es}}$ is the stored energy of the energy storage system connected to the bus n at the time step t, ϑ is the charging/discharging efficiency, Δt is the time interval, and $e_n^{\text{es,max}}$ is the capacity of the energy storage system connected to the bus n. At the initial step, the energy storage system is assumed to be fully charged, i.e., $e_{n,0}^{\text{es}}$=$e_n^{\text{es,max}}$.

• Reactive power output constraints: Eq. (37) is the constraint of the reactive power output of the SVC, and Eq. (38) is the constraint of the reactive power output of the capacitor bank.

$$q_n^{\text{svc,min}} \le q_{n,t}^{\text{svc}} \le q_n^{\text{svc,max}}, \tag{37}$$

$$q_n^{\text{cb,min}} \le q_{n,t}^{\text{cb}} \le q_n^{\text{cb,max}}, \tag{38}$$

where $q_n^{\text{svc,min}}$ and $q_n^{\text{svc,max}}$ are the minimum and maximum reactive power outputs of the SVC in the bus n, and $q_n^{\text{cb,min}}$ and $q_n^{\text{cb,max}}$ are the minimum and maximum reactive power outputs of the capacitor bank in the bus n.

3.3.2. Planning

In the planning phase, the DSO takes the actions of network reconfiguration ($\mathbf{A}_k^{\text{f}}$) and technology installation ($\mathbf{A}_k^{\text{e}}$) under the states of $\left\{\mathbf{S}_k^{\text{f}}, \mathbf{S}_k^{\text{e}}, \mathbf{S}_k^{\text{y}}\right\}$. The environment returns a cost to judge how good the DSO's actions are. This cost is defined as the sum of minimum values of the investment cost, power loss, renewable curtailment, and load curtailment. To ensure a trade-off of four objective functions, each objective value is normalised through using the min-max normalisation, before assigning equal weights. Therefore, the cost of RL can be expressed as

$$\begin{aligned}& c\left(\mathbf{S}_k^{\text{f}}, \mathbf{S}_k^{\text{e}}, \mathbf{S}_k^{\text{y}}, \mathbf{A}_k^{\text{f}}, \mathbf{A}_k^{\text{e}}\right) \\ :=& \min\left[\xi \cdot \left(\frac{c^{\text{inv}} - c^{\text{inv,min}}}{c^{\text{inv,max}} - c^{\text{inv,min}}} + \frac{p^{\text{loss}} - p^{\text{loss,min}}}{p^{\text{loss,max}} - p^{\text{loss,min}}}\right.\right. \\ & \left.\left. + \frac{\Delta p^{\text{pv}} - p^{\text{pv,min}}}{p^{\text{pv,max}} - p^{\text{pv,min}}} + \frac{\Delta p^{\text{load}} - p^{\text{load,min}}}{p^{\text{load,max}} - p^{\text{load,min}}}\right)\right],\end{aligned} \tag{39}$$

where ξ is the weight of each objective function (ξ=0.25 in the case of four objective functions with equal weights), $c^{\text{inv,min}}$ and $c^{\text{inv,max}}$ are the minimum and maximum values of the investment cost, respectively, which is obtained when $\mathbf{A}_k^{\text{e}}$ is zero matrix and matrix of ones, respectively, $p^{\text{loss,min}}$ and $p^{\text{loss,max}}$ are the minimum and maximum values of the total power loss of a distribution network, respectively, which is obtained when $\tilde{i}_{m,n,t}$ equals to 0 and $\left(i_{m,n}^{\max}\right)^2$, $\forall n, m \in \mathcal{N}, t \in \mathcal{T}$, respectively, $p^{\text{pv,min}}$ and $p^{\text{pv,max}}$ are the minimum and maximum values of the total solar power curtailment of a distribution network, respectively, which is obtained when $\Delta p_{n,t}^{\text{pv}}$ equals to 0 and $p_{n,t}^{\text{pv}}$, $\forall n \in \mathcal{N}, t \in \mathcal{T}$, respectively, and $p^{\text{load,min}}$ and $p^{\text{load,max}}$ are the minimum and maximum values of the total load curtailment of a distribution network, respectively, which is obtained when $\Delta p_{n,t}^{\text{load}}$ equals to 0 and $p_{n,t}^{\text{load,net}}$, $\forall n \in \mathcal{N}, t \in \mathcal{T}$, respectively.

3.3.3. Adaption of planning and operational decisions

For a DSO, the planning decisions include the network reconfiguration ($\mathbf{A}_k^{\text{f}}$) and technology installation ($\mathbf{A}_k^{\text{e}}$), and the

operational control decisions include the active/reactive power import ($p^{\text{im}}_{n,t}/q^{\text{im}}_{n,t}$), bus voltage ($\tilde{v}_{n,t}$), line current ($\tilde{i}_{m,n,t}$), active/reactive power flow ($p_{n,o,t}/q_{n,o,t}$ and $p_{m,n,t}/q_{m,n,t}$), charging/discharging rate of storage ($p^{\text{es}}_{n,t}$), and reactive power outputs of the SVC ($q^{\text{svc}}_{n,t}$) and capacitor bank ($q^{\text{cb}}_{n,t}$).

The designed digital twin is able to adapt both the planning and operational control decisions with transitions of a distribution system, through procedures as presented in **Fig. 5**. The details of these procedures are explained as follows:

Step 1: When a distribution system transitions to a new state, the actions ($\mathbf{A}^{\text{f}}_k$ and $\mathbf{A}^{\text{e}}_k$) are taken driven by minimising the cost of the RL as Eq. (39).

Step 2: Once actions are taken, the network configurations and technology installations would be fixed, based on which the DSO makes control decisions ($p^{\text{im}}_{n,t}$, $q^{\text{im}}_{n,t}$, $\tilde{v}_{n,t}$, $\tilde{i}_{m,n,t}$, $p_{n,o,t}$, $q_{n,o,t}$, $p_{m,n,t}$, $q_{m,n,t}$, $p^{\text{es}}_{n,t}$, $q^{\text{svc}}_{n,t}$, $q^{\text{cb}}_{n,t}$) through solving the optimisation problem in Eq. (39), being subject to constraints of Eqs. (25) - (38).

Step 3: The sum of optimal values of four objective functions are taken as the cost of the RL to judge how good the actions are.

This procedures are iteratively proceeded through training the RL model as detailed in the next section, until finding the optimal solutions for both planning and operational control decisions.

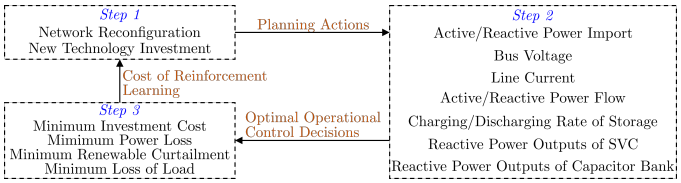

Fig. 5. Schematic illustration for procedures of adapting planning and operational control decisions with transitions of a distribution system.

Remark: For practical implementations, the time step corresponds to the real-time operation of distribution systems. For instance, for a half-hour time interval of the daily operation, we have ($\Delta t, |\mathcal{T}|$) = (0.5,48) and t=1,2,...,48. The stage corresponds to the long-term transitions of distribution systems. For instance, the composition of consumers' load patterns changes after 15 days, the stage interval Δk = (15×24) h.

4. Batch reinforcement learning

This section discusses the proposed data pre-processing algorithm, upon which the FQI is developed to predict the optimal Q-function with the states and actions as the inputs, through iteratively training the CNNs. The off-policy RL algorithms, e.g., Q-learning, FQI, and deep Q-network, have the potential to make full use of historical data, compared to the on-policy algorithms, e.g., SARSA. Given the action spaces of network reconfiguration and new technology investment are not continuous, the Q-learning, FQI, and deep Q-network are advantageous compared to the deep deterministic policy gradient. The FQI uses a batch of transitions to iteratively update the Q-value estimates and uses neural networks to handle high-dimensional input spaces, which improves from the traditional Q-learning. Hence, this research uses the FQI as a batch RL approach.

4.1. Data pre-processing

Initial network parameters (i.e., weights and bias) are not necessary before the data pre-processing step, since the step of data pre-processing itself serves as an initialisation of network parameters. The outputs of the data pre-processing are batches of tuples which will be used in the initial step of the FQI to train the initial network parameters. These tuples include: *(1)* current states $\mathbf{S}^{\text{f}}_k, \mathbf{S}^{\text{e}}_k, \mathbf{S}^{\text{y}}_k$, *(2)* current actions $\mathbf{A}^{\text{f}}_k, \mathbf{A}^{\text{e}}_k$, *(3)* next states $\mathbf{S}^{\text{f}}_{k+1}, \mathbf{S}^{\text{e}}_{k+1}, \mathbf{S}^{\text{y}}_{k+1}$, and *(4)* cost $c\left(\mathbf{S}^{\text{f}}_k, \mathbf{S}^{\text{e}}_k, \mathbf{S}^{\text{y}}_k, \mathbf{A}^{\text{f}}_k, \mathbf{A}^{\text{e}}_k\right)$. Let b denote the index of a batch and $\mathcal{B}$ is the set of batches. Each batch contains $|\mathcal{K}|$ stages. Hence, the data pre-processing would generate $|\mathcal{B}| \times |\mathcal{K}|$ tuples as $\left\{\mathbf{S}^{\text{f}}_k, \mathbf{S}^{\text{e}}_k, \mathbf{S}^{\text{y}}_k, \mathbf{A}^{\text{f}}_k, \mathbf{A}^{\text{e}}_k, \mathbf{S}^{\text{f}}_{k+1}, \mathbf{S}^{\text{e}}_{k+1}, \mathbf{S}^{\text{y}}_{k+1}, c\left(\mathbf{S}^{\text{f}}_k, \mathbf{S}^{\text{e}}_k, \mathbf{S}^{\text{y}}_k, \mathbf{A}^{\text{f}}_k, \mathbf{A}^{\text{e}}_k\right)\right\}, \forall k = 1, ..., |\mathcal{B}| \times |\mathcal{K}|$. The inputs of the data pre-processing are the states of the network configuration and technology installation at the initial stage, denoted as $\mathbf{S}^{\text{f}}_0$ and $\mathbf{S}^{\text{e}}_0$, respectively, and the exogenous state of the load pattern over the entire $|\mathcal{B}| \times |\mathcal{K}|$ stages, $\mathbf{S}^{\text{y}}_k, \forall k = 1, ..., |\mathcal{B}| \times |\mathcal{K}|$. For each iteration k, the procedures of the pre-training are detailed as follows:

Step 1: Randomly generate the matrices of the actions of the network reconfiguration, i.e., $\mathbf{A}^{\text{f}}_k$, and technology investment, i.e., $\mathbf{A}^{\text{e}}_k$, being subject to the constraint of Eq. (19).

Step 2: Calculate the next states of the network configuration, i.e., $\mathbf{S}^{\text{f}}_{k+1}$, and technology installation, i.e., $\mathbf{S}^{\text{e}}_{k+1}$, according to Eq. (18) and Eq. (20), respectively.

Step 3: Calculate the cost, i.e., $c\left(\mathbf{S}^{\text{f}}_k, \mathbf{S}^{\text{e}}_k, \mathbf{S}^{\text{y}}_k, \mathbf{A}^{\text{f}}_k, \mathbf{A}^{\text{e}}_k\right)$, by solving the multi-objective optimisation problem of Eq. (39), being subject to the constraints of Eq. (25) - Eq. (38).

The algorithm of the data pre-processing is summarised in **Algorithm 1**.

Algorithm 1 Algorithm of the data pre-processing

input: $\mathbf{S}^{\text{f}}_0, \mathbf{S}^{\text{e}}_0, \mathbf{S}^{\text{y}}_k$

1: **for** $k = 1, ..., |\mathcal{B}| \times |\mathcal{K}|$ **do**
2: randomly generate $\mathbf{A}^{\text{f}}_k$ and $\mathbf{A}^{\text{e}}_k$, being subject to Eq. (19)
3: calculate $\mathbf{S}^{\text{f}}_{k+1}$ and $\mathbf{S}^{\text{e}}_{k+1}$, according to Eq. (18) and Eq. (20), respectively
4: calculate $c\left(\mathbf{S}^{\text{f}}_k, \mathbf{S}^{\text{e}}_k, \mathbf{S}^{\text{y}}_k, \mathbf{A}^{\text{f}}_k, \mathbf{A}^{\text{e}}_k\right)$ by solving Eq. (39), being subject to the constraints of Eq. (25) - Eq. (38).
5: **end for**

output: $\left\{\mathbf{S}^{\text{f}}_k, \mathbf{S}^{\text{e}}_k, \mathbf{S}^{\text{y}}_k, \mathbf{A}^{\text{f}}_k, \mathbf{A}^{\text{e}}_k, \mathbf{S}^{\text{f}}_{k+1}, \mathbf{S}^{\text{e}}_{k+1}, \mathbf{S}^{\text{y}}_{k+1}, c\left(\mathbf{S}^{\text{f}}_k, \mathbf{S}^{\text{e}}_k, \mathbf{S}^{\text{y}}_k, \mathbf{A}^{\text{f}}_k, \mathbf{A}^{\text{e}}_k\right)\right\}$

4.2. Convolutional neural networks

The CNNs are used to predict the optimal Q-function with the inputs of the states and actions as

$$\hat{q}^* = f^{\text{NN}}\left(\mathbf{S}^{\text{f}}_k, \mathbf{S}^{\text{e}}_k, \mathbf{S}^{\text{y}}_k, \mathbf{A}^{\text{f}}_k, \mathbf{A}^{\text{e}}_k\right), \tag{40}$$

where $\hat{q}^*$ is the predicted optimal Q-function, and $f^{\text{NN}}(\cdot)$ is the regression function parametrised by tuning the CNNs.

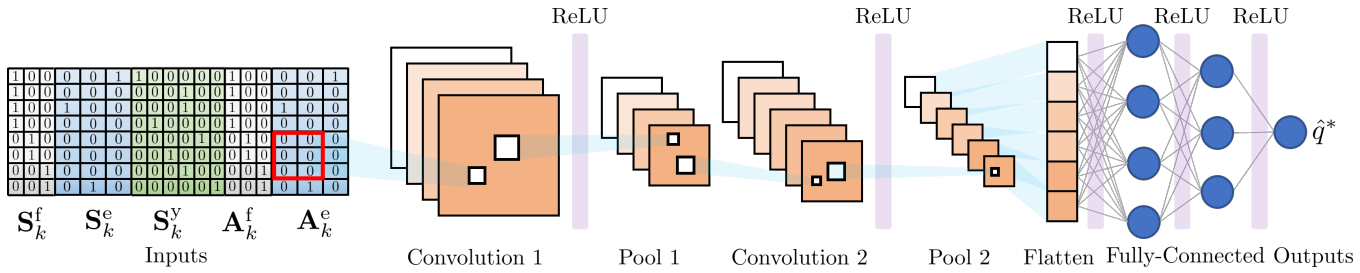

Fig. 6. Architecture of the designed convolutional neural networks. The inputs are matrices of states and actions, and outputs are the predicted optimal Q-functions. Each convolutional layer is followed by a pooling layer. The filter (red box) slides through the input matrices to extract feature representations as a feature map. The global feature map is flatten as a vector and further processed by fully-connected layers.

The architecture of the proposed CNNs is presented in **Fig. 6**. The CNNs are capable of extracting feature representations from high-dimensional inputs, i.e., the matrices of states and actions. It is especially useful for capturing the spatial features of these matrices, i.e., how the network configuration, technology installation, and load pattern in one bus are related to its surrounding buses within the filter size of the CNNs (see the red box in **Fig. 6**). Each filter slides through the input matrices to extract feature representations for generating a feature map as

$$\Phi = f^{\mathrm{ReLU}}\left[\mathbf{W} \otimes \left(\mathbf{S}_k^{\mathrm{f}}, \mathbf{S}_k^{\mathrm{e}}, \mathbf{S}_k^{\mathrm{y}}, \mathbf{A}_k^{\mathrm{f}}, \mathbf{A}_k^{\mathrm{e}}\right) + \varrho\right], \tag{41}$$

where Φ is the feature map generated by a filter, $f^{\mathrm{ReLU}}(\cdot)$ is the activation function, e.g., rectified linear unit (ReLU), $\mathbf{W}$ is the weight matrix of a filter, and ϱ is the bias term.

Each convolutional layer is followed by a pooling layer to reduce the spatial size of extracted features while keep key features. The same padding is used to pad the input of each convolutional layer and pooling layer with zeros around the border, in order to keep the input size fitting the filter size. Multiple feature maps generated by various filters are finally stacked as a global feature map, before being flatten as a vector and further processed by fully-connected layers.

4.3. Fitted Q-iteration

The aim of the FQI is to predict the optimal Q-function through iteratively training neural networks with the transition of states. With the predicted optimal Q-function, the actions for the network reconfiguration and technology investment can be determined according to Eq. (6). The inputs of the FQI are batches of tuples $\left\{\mathbf{S}_k^{\mathrm{f}}, \mathbf{S}_k^{\mathrm{e}}, \mathbf{S}_k^{\mathrm{y}}, \mathbf{A}_k^{\mathrm{f}}, \mathbf{A}_k^{\mathrm{e}}, \mathbf{S}_{k+1}^{\mathrm{f}}, \mathbf{S}_{k+1}^{\mathrm{e}}, \mathbf{S}_{k+1}^{\mathrm{y}}, c\left(\mathbf{S}_k^{\mathrm{f}}, \mathbf{S}_k^{\mathrm{e}}, \mathbf{S}_k^{\mathrm{y}}, \mathbf{A}_k^{\mathrm{f}}, \mathbf{A}_k^{\mathrm{e}}\right)\right\}, \forall k = 1, ..., |\mathcal{B}| \times |\mathcal{K}|$ generated by **Algorithm 1**. The procedures of the FQI are detailed as follows:

Step 1: At the initial stage, there is no trained neural networks to predict the optimal Q-function. Hence, the minimum value of the optimal Q-function is assumed as 0. According to the Bellman optimality equation in Eq. (5), $q^*(\mathbf{S}_k, \mathbf{A}_k) = c(\mathbf{S}_k, \mathbf{A}_k)$. Hence, train the initialised neural networks f_0^{NN} with $|\mathcal{B}| \times |\mathcal{K}|$ states and actions, i.e., $\mathbf{S}_k^{\mathrm{f}}, \mathbf{S}_k^{\mathrm{e}}, \mathbf{S}_k^{\mathrm{y}}, \mathbf{A}_k^{\mathrm{f}}, \mathbf{A}_k^{\mathrm{e}}, \forall k = 1, ..., |\mathcal{B}| \times |\mathcal{K}|$, as inputs, and $|\mathcal{B}| \times |\mathcal{K}|$ costs, i.e., $c\left(\mathbf{S}_k^{\mathrm{f}}, \mathbf{S}_k^{\mathrm{e}}, \mathbf{S}_k^{\mathrm{y}}, \mathbf{A}_k^{\mathrm{f}}, \mathbf{A}_k^{\mathrm{e}}\right), \forall k = 1, ..., |\mathcal{B}| \times |\mathcal{K}|$, as outputs.

Step 2: Take each batch b as an iteration. For each stage k within the iteration b, given the next states, i.e., $\mathbf{S}_{k+1}^{\mathrm{f}}, \mathbf{S}_{k+1}^{\mathrm{e}}, \mathbf{S}_{k+1}^{\mathrm{y}}$, use the trained neural networks from the last iteration, i.e., f_{b-1}^{NN}, to predict the optimal Q-function, and find the minimum optimal Q-function through heuristically searching the optimal actions for the next stage, i.e., $\mathbf{A}_{k+1}^{\mathrm{f}}, \mathbf{A}_{k+1}^{\mathrm{e}}$, as

$$\min_{\mathbf{A}_{k+1}^{\mathrm{f}}, \mathbf{A}_{k+1}^{\mathrm{e}}} \hat{q}^* = f_{b-1}^{\mathrm{NN}}\left(\mathbf{S}_{k+1}^{\mathrm{f}}, \mathbf{S}_{k+1}^{\mathrm{e}}, \mathbf{S}_{k+1}^{\mathrm{y}}, \mathbf{A}_{k+1}^{\mathrm{f}}, \mathbf{A}_{k+1}^{\mathrm{e}}\right). \tag{42}$$

According to Eq. (6), the corresponding optimal actions for the stage $k+1$ can be obtained as

$$\mathbf{A}_{k+1}^{\mathrm{f}}, \mathbf{A}_{k+1}^{\mathrm{e}} = \arg \min_{\mathbf{A}_{k+1}^{\mathrm{f}}, \mathbf{A}_{k+1}^{\mathrm{e}}} \hat{q}^* \tag{43}$$

Step 3: Calculate the optimal Q-function of the stage k according to the Bellman optimality equation in Eq. (5) as

$$q^*\left(\mathbf{S}_k^{\mathrm{f}}, \mathbf{S}_k^{\mathrm{e}}, \mathbf{S}_k^{\mathrm{y}}, \mathbf{A}_k^{\mathrm{f}}, \mathbf{A}_k^{\mathrm{e}}\right) = c\left(\mathbf{S}_k^{\mathrm{f}}, \mathbf{S}_k^{\mathrm{e}}, \mathbf{S}_k^{\mathrm{y}}, \mathbf{A}_k^{\mathrm{f}}, \mathbf{A}_k^{\mathrm{e}}\right) + \min_{\mathbf{A}_{k+1}^{\mathrm{f}}, \mathbf{A}_{k+1}^{\mathrm{e}}} \hat{q}^*. \tag{44}$$

Step 4: Once the optimal Q-function for the entire $|\mathcal{K}|$ stages are obtained, use $|\mathcal{K}|$ states and actions, i.e., $\mathbf{S}_k^{\mathrm{f}}, \mathbf{S}_k^{\mathrm{e}}, \mathbf{S}_k^{\mathrm{y}}, \mathbf{A}_k^{\mathrm{f}}, \mathbf{A}_k^{\mathrm{e}}, \forall k = 1, ..., |\mathcal{K}|$, as inputs, and $|\mathcal{K}|$ optimal Q-functions, i.e., $q^*\left(\mathbf{S}_k^{\mathrm{f}}, \mathbf{S}_k^{\mathrm{e}}, \mathbf{S}_k^{\mathrm{y}}, \mathbf{A}_k^{\mathrm{f}}, \mathbf{A}_k^{\mathrm{e}}\right), \forall k = 1, ..., |\mathcal{K}|$, as outputs to train the neural networks f_b^{NN}.

The algorithm of the FQI is summarised in **Algorithm 2**.

Algorithm 2 Algorithm of the fitted Q-iteration

input: $\left\{\mathbf{S}_k^{\mathrm{f}}, \mathbf{S}_k^{\mathrm{e}}, \mathbf{S}_k^{\mathrm{y}}, \mathbf{A}_k^{\mathrm{f}}, \mathbf{A}_k^{\mathrm{e}}, \mathbf{S}_{k+1}^{\mathrm{f}}, \mathbf{S}_{k+1}^{\mathrm{e}}, \mathbf{S}_{k+1}^{\mathrm{y}}, c\left(\mathbf{S}_k^{\mathrm{f}}, \mathbf{S}_k^{\mathrm{e}}, \mathbf{S}_k^{\mathrm{y}}, \mathbf{A}_k^{\mathrm{f}}, \mathbf{A}_k^{\mathrm{e}}\right)\right\}$

train f_0^{NN} with $\mathbf{S}_k^{\mathrm{f}}, \mathbf{S}_k^{\mathrm{e}}, \mathbf{S}_k^{\mathrm{y}}, \mathbf{A}_k^{\mathrm{f}}, \mathbf{A}_k^{\mathrm{e}}$ as inputs and $c\left(\mathbf{S}_k^{\mathrm{f}}, \mathbf{S}_k^{\mathrm{e}}, \mathbf{S}_k^{\mathrm{y}}, \mathbf{A}_k^{\mathrm{f}}, \mathbf{A}_k^{\mathrm{e}}\right)$ as outputs
for $b = 1, ..., |\mathcal{B}|$ **do**
  for $k = 1, ..., |\mathcal{K}|$ **do**
    use f_{b-1}^{NN} to predict $\hat{q}^*$ and find the minimum $\hat{q}^*$ through heuristically searching $\mathbf{A}_{k+1}^{\mathrm{f}}, \mathbf{A}_{k+1}^{\mathrm{e}}$ according to Eq. (42)
    calculate $q^*\left(\mathbf{S}_k^{\mathrm{f}}, \mathbf{S}_k^{\mathrm{e}}, \mathbf{S}_k^{\mathrm{y}}, \mathbf{A}_k^{\mathrm{f}}, \mathbf{A}_k^{\mathrm{e}}\right)$ according to Eq. (44)
  end for
  train f_b^{NN} with $\mathbf{S}_k^{\mathrm{f}}, \mathbf{S}_k^{\mathrm{e}}, \mathbf{S}_k^{\mathrm{y}}, \mathbf{A}_k^{\mathrm{f}}, \mathbf{A}_k^{\mathrm{e}}$ as inputs and $q^*\left(\mathbf{S}_k^{\mathrm{f}}, \mathbf{S}_k^{\mathrm{e}}, \mathbf{S}_k^{\mathrm{y}}, \mathbf{A}_k^{\mathrm{f}}, \mathbf{A}_k^{\mathrm{e}}\right)$ as outputs
end for

output: trained f_b^{NN}

The Flowchart of the batch RL and interactions of the data pre-processing, neural network training, and FQI is illustrated in **Fig. 7**.

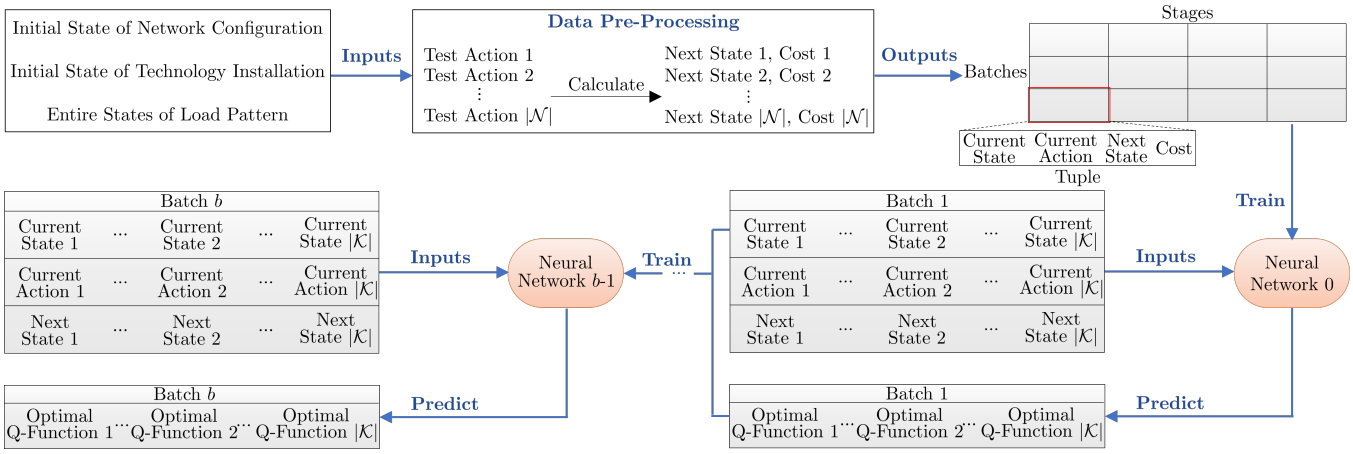

Fig. 7. Flowchart of the batch reinforcement learning and interactions of the data pre-processing, neural network training, and fitted Q-iteration.

5. Case studies

Case studies have been conducted to validate the performances of the proposed model in terms of the learning accuracy, planning and operational strategies, and model scalability.

5.1. Simulation setup

The simulations were performed on a machine with Intel(R) Core(TM) i7-9700K CPU @ 3.00GHz and a NVIDIA GeForce RTX 2080 GPU. The proposed model was written in the Python, through which the neural networks were implemented by Pytorch [29] and power flow analysis was implemented by Pandapower [30]. The optimisation problems in this study were solved by Gurobi Optimizer [31]. The simulation was primarily implemented on the IEEE 33-bus distribution network and compared to the IEEE 18-bus distribution network and IEEE 69-bus distribution network to evaluate the scalability.

The consumption data was sourced from 55 loads in the GB low voltage distribution networks [32] in every half-hour over one year. In each season of this year, 6 typical load patterns and their occurrence probabilities were firstly generated through using the proposed CPM approach, in which 3 load patterns were randomly selected to be integrated with the future energy scenarios. The PV data was sourced from [33] and electric vehicle charging profiles were sourced from [34]. The fitted Q-iteration was performed when the system transitions from one season to another, i.e., batches $|\mathcal{B}| = 4$. Other parameters used in the simulation are listed in **Table 2**.

The 6 typical load patterns were arbitrarily assigned to the metering points representing the loads of the distribution network by the proportion of occurrence probabilities. This arbitrary assignment was performed 500 times to create 500 states of load patterns, i.e., $|\mathcal{K}| = 500$. The action, next state, and cost are calculated by the proposed data pre-processing algorithm, forming 500 tuples. These 500 tuples are not sequential and therefore were randomly split into 80% of the training set and 20% of the testing set. Good hyper-parameters would improve the learning accuracy and computational efficiency. The z-score normalisation was used to normalise the batches as the inputs of the CNNs. The Adam Optimiser [35] was used to train the CNNs for 50 epochs, with 1×10^{-2} initial learning rate and 1×10^{-2} weight decay of the L2 regularisation [36]. The scheduler was used to adjust the learning rate, through which the learning rate would be reduced by 1×10^{-2} if there was no improvement of the testing accuracy for continuous 5 epochs. 1×10^{-2} dropout [37] was used to randomly drop units of the CNNs in avoiding the issue of overfitting. The parameters of the designed CNNs are shown in **Table 3**.

Table 2 Parameters used in the case studies.

Parameter	Value	Parameter	Value	Parameter	Value
$p^{\mathrm{tr,min}}$	-500 kW	$p^{\mathrm{tr,max}}$	500 kW	$q^{\mathrm{tr,min}}$	-500 kVar
$q^{\mathrm{tr,max}}$	500 kVar	$v_n^{\min}$	0.95 pu	$v_n^{\max}$	1.05 pu
$i_{m,n}^{\max}$	0.400 kA	$p_n^{\mathrm{es,max}}$	300 kW	$e_{n,t}^{\mathrm{es}}$	1000 kWh
$q_n^{\mathrm{svc,min}}$	- 500 kVar	$q_n^{\mathrm{svc,max}}$	500 kVar	$q_n^{\mathrm{cb,min}}$	0 kVar
$q_n^{\mathrm{cb,max}}$	500 kVar	ϑ	95%	r^{u}	0.7 Ω/km
x^{u}	0.7 Ω/km	μ	409.09 W/K	ν	1.75×10^6 J/kgK
p_t^{hp}	10 kW	θ	4	$\tau^{\min}$	18 °C

Table 3 Parameters of the designed convolutional neural networks for the IEEE 33-bus distribution system.

Layer	Input size	Output size	Filter number	Filter size	Stride	Padding
Convolutional 1	33,18,1	33,18,32	32	5×5	1	2
Pooling 1	33,18,32	16,9,32	1	2×2	2	0
Convolution 2	16,9,32	16,9,64	64	3×3	1	1
Pooling 2	16,9,64	8,4,64	1	2×2	2	0
Flatten	8,4,64	2048	-	-	-	-
Fully-connected 1	2048	1024	-	-	-	-
Fully-connected 2	1024	512	-	-	-	-
Fully-connected 3	512	256	-	-	-	-
Output	256	1	-	-	-	-

5.2. Evaluation of learning accuracy

The learning losses are measured by the mean squared error (MSE) [38] as

$$\mathrm{MSE} := \frac{1}{|\mathcal{K}|}\sum_{k\in\mathcal{K}}\left[q^*\left(\mathbf{S}_k^{\mathrm{f}},\mathbf{S}_k^{\mathrm{e}},\mathbf{S}_k^{\mathrm{y}},\mathbf{A}_k^{\mathrm{f}},\mathbf{A}_k^{\mathrm{e}}\right)-\hat{q}^*\left(\mathbf{S}_k^{\mathrm{f}},\mathbf{S}_k^{\mathrm{e}},\mathbf{S}_k^{\mathrm{y}},\mathbf{A}_k^{\mathrm{f}},\mathbf{A}_k^{\mathrm{e}}\right)\right]. \tag{45}$$

To evaluate the accuracy and convergence on the training and testing, the designed CNNs are compared with the deep neural networks (DNNs), long short-term memory (LSTM) of recurrent neural networks [39], and transformer networks [40] with the differences as:

• *DNNs:* Two convolutional layers of our designed CNNs are replaced by two 1024-node hidden layers.

• *LSTM:* Two convolutional layers of our designed CNNs are replaced by two 1024-node hidden layers of LSTM. The LSTM imports each row of the input matrix at each time step and returns a 1024 vector as a global memory after processing the entire matrix, i.e., 33 time steps.

• *Transformer Networks:* The convolutional layers of our designed CNNs are replaced by the transformer architecture (with a two-layer encoder only) with the setting of a 1024 feed-forward dimension, 6 attention heads and positional encoding.

Other settings and parameters are remained the same as our designed CNNs. If the scheduler reduces the learning rate below 1×10^{-6} and no improvement of the accuracy is observed for continuous 5 epochs, the learning process is defined as the convergence. The losses of training and testing for 4 iterations of the FQI are shown in **Fig. 8**. The computational time for the CNNs, DNNs, LSTM, and transformer networks are 12,246 s, 10,036 s, 38,752 s, and 55,840 s respectively. It can be seen that both the training and testing of four neural networks converge within 50 epochs. The CNNs, LSTM, and transformer networks yield better training and testing performances compared to the DNNs. This is because the CNNs are able to extract spatial features from high-dimensional input matrices; The LSTM of recurrent neural networks are able to store key features into the hidden memory when processing each row of input matrices; The transformer networks are able to pay equal attention to all the elements of input matrices and understand relationships of these elements through using the attention mechanism. Nonetheless, the CNNs save 68.40% of computational time compared to the LSTM and 78.07% of computational time compared to the transformer networks. This is because storing memories and processing the high-dimensional matrices by each row cause additional computational burdens for LSTM, and the computational complexity of the transformer networks quadratically increases with the sequence length of inputs.

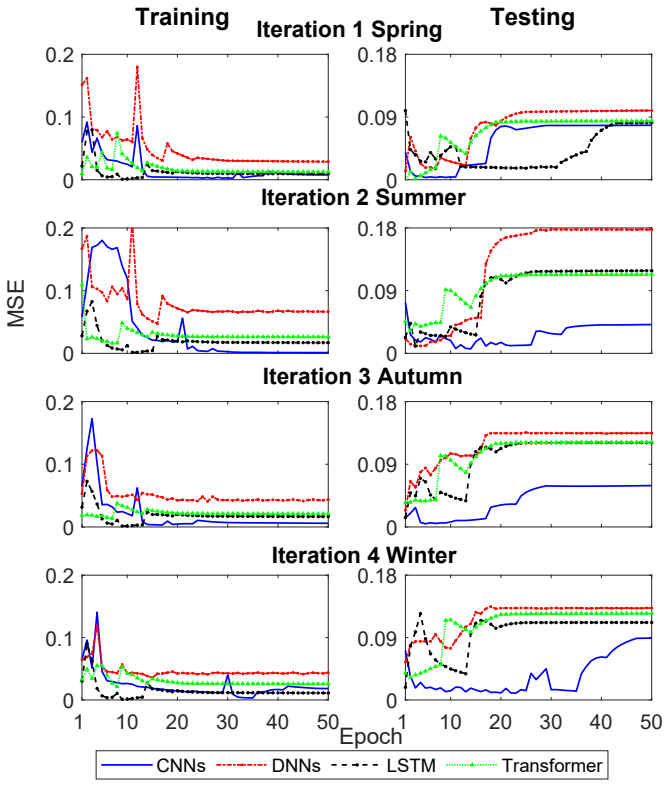

Fig. 8. Evaluation of the training and testing accuracy for our designed convolutional neural networks (CNNs), deep neural networks (DNNs), long short-term memory (LSTM) of recurrent neural networks, and transformer networks. The x axes indicate the learning epoch, and y axes indicate the loss of the mean squared error (MSE).

5.3. Evaluation of model performances

First, the fitting results of the kernel density estimation is presented in **Fig. 9**. The time steps 8, 16, 24, 32, 40, and 48 in winter are sampled to illustrate the fitting performance. The results show a well fitted density function for every time step with 8.82×10^{-7} of the average MSE. Based on the fitted density function, typical load patterns in the summer and winter are sampled as shown in **Fig. 10** and **Fig. 11**, respectively. The red lines are typical clusters and the grey lines are original power profiles belonging to each cluster (sampled five of them for each cluster for illustrating clarity). It can be seen from the figures that there are some high consumptions in certain time of the winter and their power demands are almost doubled compared to those in the summer. This is caused by the electrification of heat, i.e., the integration of air-source heat pumps in future energy scenarios.

Second, to evaluate the performances of our proposed model in terms of *(1)* investment cost, *(2)* power loss, *(3)* loss of load, and *(4)* renewable curtailment, the following four cases are compared:

• *Case 1 (Benchmark optimisation):* Instead of predicting the optimal Q-function using the CNNs as in our proposed model, the benchmark optimisation finds the optimal Q-function through solving the optimisation problem, which yields a theoretical benchmark to evaluate how far the predicted optimal decisions of our proposed model are from the theoretical optimal ones.

• *Case 2 (Single-stage optimisation):* Instead of iteratively adapting the CNNs with the dynamic state transitions as in the FQI, the single-stage optimisation is a model-based solution which solves the optimisation problems in Eq. (39) for every stage independently.

• *Case 3 (Proposed model):* Our proposed model iteratively trains the CNNs through the FQI algorithm. Optimal actions can be yielded from the predicted optimal Q-function.

• *Case 4 (Monte Carlo tree search):* As the approach developed in our previous research [14], the Monte Carlo tree search based RL is used to find optimal actions by simulating many possible trajectories and choosing the best one.

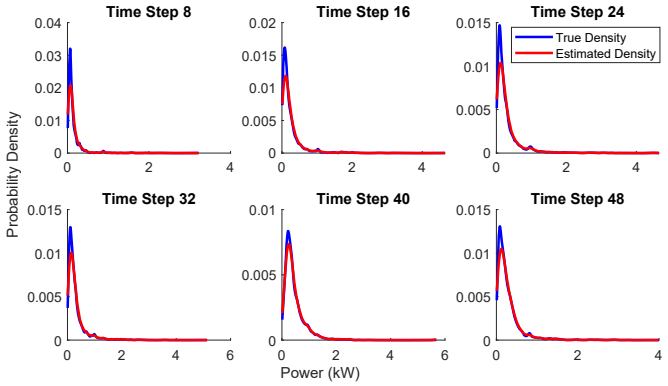

Fig. 9. Fitting results of the kernel density estimation. The time steps 8, 16, 24, 32, 40, and 48 in winter are sampled to illustrate the fitting performance. The x axes indicate the range of power distribution, and the y axes indicate the probability density function.

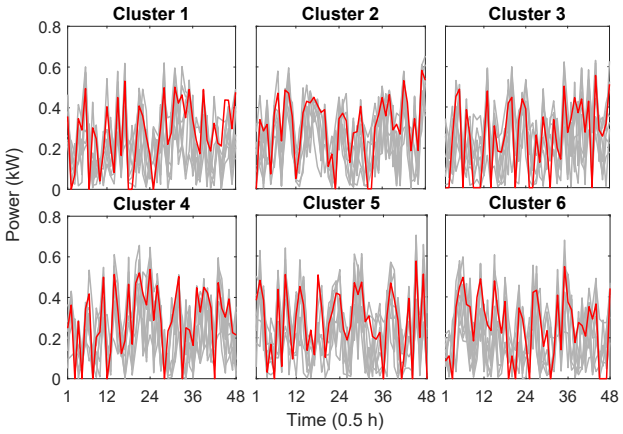

Fig. 10. Typical load patterns in summer. The red lines are typical clusters and the grey lines are original power profiles belonging to each cluster (sampled five of them for each cluster for illustrating clarity). The x axes indicate the time steps, and the y axes indicate the power demand.

• *Case 5 (Deep deterministic policy gradient [41]):* The deep deterministic policy gradient is also used as a comparison, in which the critic and critic target networks have the same structure as our designed CNNs, and the actor and actor target networks consist of 2 hidden layers with the node numbers of 400 and 300. The discount factor is set as 0.99, and the soft update coefficient is set as 0.001.

The comparison of our proposed model, single-stage optimisation, Monte Carlo tree search, and deep deterministic policy gradient against the benchmark optimisation in terms of the investment cost, power loss, loss of load, and renewable curtailment is presented in **Fig. 12**. It can be seen that through iteratively training the neural networks with the system transitions, the results of our proposed model in the four criteria are closest to the theoretical optimal results yielded by the benchmark optimisation (lower percentage levels compared to the single-stage optimisation, Monte Carlo tree search, and deep deterministic policy gradient). The investment costs of our proposed model are reduced by approximately 50% when the system transitions towards winter, compared to the single-stage optimisation (increased by approximately 3 times). This is because the proposed model considers the cumulative costs covering multiple stages, and adapt the model with the dynamic system transitions. In addition, the power loss and loss of load of our proposed model, Monte Carlo tree search, and deep deterministic policy gradient can be maintained to a negligible level (within 5% compared to the benchmark optimisation) through optimal operational control and planning on the network structure and technology installation. For the renewable curtailment, the RL algorithms pay more attention to the security of supply, so as to maintain higher levels for the state of charge of storages. After the storages are fully charged, the rest renewable generation is curtailed. By contrast, the single-stage optimisation has to curtail loads on the occasion when the storage discharge and renewable generation are unable to meet the demand. The percentage of the loss of load in the single-stage optimisation is also higher than that of the renewable curtailment in the RL algorithms. On the context of adapting seasonal transition, our proposed model is less sensitive to the variance in the estimates of the Q-values, and therefore yields better performances than the Monte Carlo tree search. Compared to the deep deterministic policy gradient, the FQI is more suitable for discrete actions and simple for implementations through a straightforward iterative process of training CNNs.

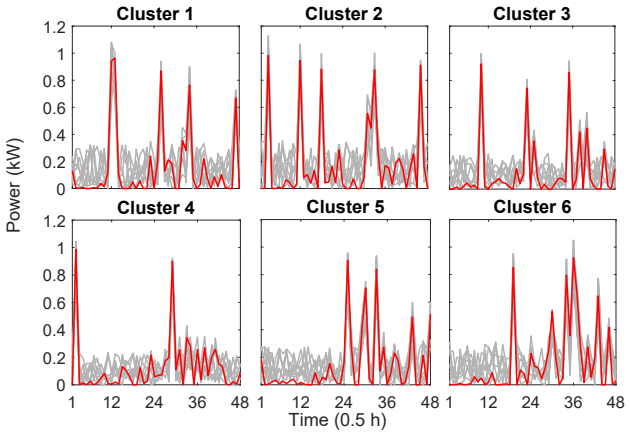

Fig. 11. Typical load patterns in winter. The red lines are typical clusters and the grey lines are original power profiles belonging to each cluster (sampled five of them for each cluster for illustrating clarity). The x axes indicate the time steps, and the y axes indicate the power demand.

Third, to illustrate how our proposed model assists the DSO to adapt optimal planning decisions with the system transitions, responding to the transition of load pattern from summer (see **Fig. 10**) to winter (see **Fig. 11**), the transitions of network configuration and technology installation are shown in **Fig. 13**. Facing the transition of the load patterns, the DSO determines the reconfiguration of the distribution network and invests in new technologies to balance the active and reactive power in the distribution network, with the targets of minimising the investment cost, power loss, loss of load, and renewable curtailment.

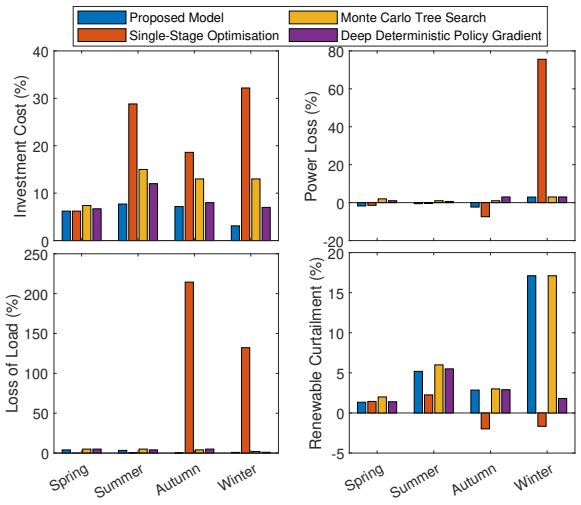

Fig. 12. Evaluation of the investment cost, power loss, loss of load, and renewable curtailment for our proposed model, single-stage optimisation, Monte Carlo tree search, and deep deterministic policy gradient against the benchmark optimisation. The x axes indicate the system stages (seasons). The y axes indicate the percentage change against the benchmark optimisation.

5.4. Evaluation of scalability

To evaluate the scalability of our proposed model, the IEEE 18-bus distribution system, IEEE 33-bus distribution system, and IEEE 69-bus distribution system were used to test the computational time in comparison with the benchmark optimisation. The parameters of CNNs for the IEEE 18-bus distribution system and IEEE 69-bus distribution system are shown in **Table 4**. The filter number, filter size, stride, padding, and parameters of the fully-connected and output layers are the same as listed in **Table 3**. The comparison of computational time between the proposed model and benchmark optimisation is shown in **Table 5**. The computational time for our proposed model includes the time used for the data pre-processing, iteratively training the CNNs using the FQI, and predicting the optimal Q-functions by trained CNNs. The computational time for the benchmark optimisation includes the time used for the data pre-processing and iteratively solving the optimisation problem to obtain the optimal Q-function. Since the data pre-processing used in both the proposed model and benchmark optimisation is identical, the only difference of computational time is caused by the difference between training CNNs and solving the optimisation problem. Once the CNNs are well trained, the optimal decisions can be produced in microseconds. As indicated in the table, compared to the benchmark optimisation, our proposed model can reduce the computational time through using the trained CNNs to process the high-dimensional inputs of states and actions and predict optimal Q-functions. It is in particular when the scale of the distribution system increases, the computational time of our proposed model remains almost unchanged whereas the computational time of the benchmark optimisation dramatically increases. Therefore, our proposed model saves 68.18%, 82.89%, and 90.81% of computational time from the benchmark optimisation under the IEEE 18-bus, 33-bus, and 69-bus distribution networks, respectively.

Table 4 Parameters of the designed convolutional neural networks for the IEEE 18-bus distribution system and IEEE 69-bus distribution system.

Layer	IEEE 18-bus distribution system		IEEE 69-bus distribution system	
	Input size	Output size	Input size	Output size
Convolutional 1	18,20,1	18,20,32	69,28,1	69,28,32
Pooling 1	18,20,32	9,10,32	69,28,32	34,14,32
Convolution 2	9,10,32	9,10,64	34,14,32	34,14,64
Pooling 2	9,10,64	4,5,64	34,14,64	17,7,64
Flatten	4,5,64	1280	17,7,64	7616

Table 5 Evaluation of the scalability on computational time under various IEEE distribution networks.

Computational time (s)	Proposed model	Benchmark optimisation
IEEE 18-bus distribution network	12,218	38,402
IEEE 33-bus distribution network	12,246	71,578
IEEE 69-bus distribution network	13,007	141,585

6. Research advantages, challenges, and prospects

This study digitalises physical distribution networks in supporting the planning and operational decision making of the DSO. For the predicability, on one hand, the proposed data preprocessing approach evaluates every potential scenario of planning and operation prior to the actual installation of a distribution system. On the other hand, considering future energy scenarios on the roof-top solar panels, air-source heat pumps, and electric vehicles helps envisage the potential impacts of transitions of local energy systems. For the responsiveness, once the distribution system transitions to a new state, the optimal control decisions can be taken and the operational constraints can be maintained in a real-time manner. For the interoperability, the designed digital twins are able to align the interests of stakeholders to system benefits, i.e., reducing the loss of load for consumers, avoiding renewable curtailment for distributed generators, and minimising the investment cost and power loss for the DSO. For the automation, digital twins optimise every single element of a distribution system, e.g., power flow and voltage, and directly perform those control actions.

Nonetheless, dating back to nationalised and municipal power systems, legacy assets of less monitored low voltage distribution networks still pose a challenge to this study. Overcoming this challenge requires *(1)* collection and access of topographic, demographic, social-economic, and technical information of the last mile of distribution networks, and *(2)* sophisticated model to automatically synthesise approximations of the last mile of distribution networks.

As future prospects, first, research community and industrial practise need to develop novel explainable artificial intelligence and user friendly interface, so that the model outputs can be understood by both engineers and data scientists. Second, planning and policy experts need to validate the model against envisaged system transitions, in order to make informed decisions based on the model outputs. Third, given the massive and active engagement of consumers holds the key to the provision of last mile information, proper incentive strategy needs to be developed to facilitate such engagement.

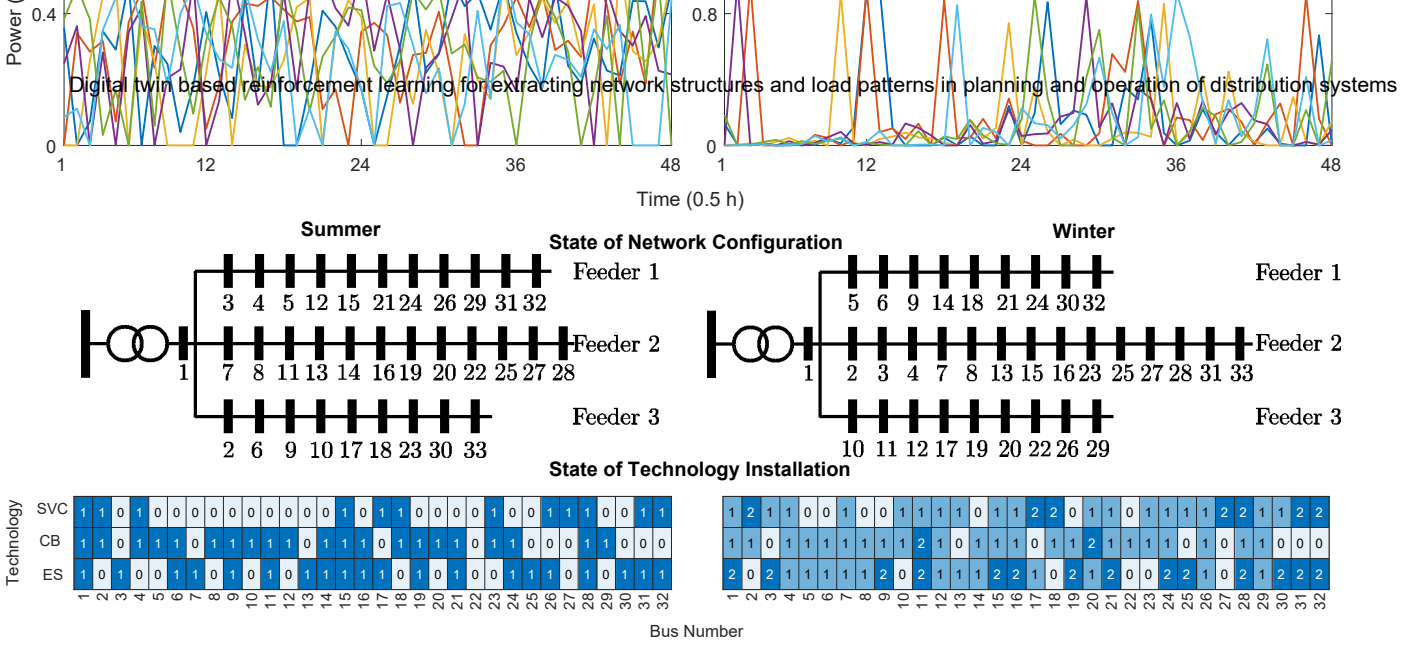

Fig. 13. Schematic illustration of the system states of the network configuration and technology installation transitioning from the summer to winter. The top two sub-figures are the state of the network configuration. The bottom two sub-figures are the state of the technology installation, in which each element indicates the total number of installed technologies.

7. Conclusions

Transitioning to low carbon power networks carries considerable technical risks. To mitigate resulting operational and financial consequences, in future, DSOs will need tools to support decision making based on realistic representations of their infrastructures. This paper proposes a digital twin based RL model to exploit data from distribution systems in supporting the decision making of the DSOs. The digital twin imports the states of network configurations, technology investments, and load patterns from physical distribution systems. These high-dimensional states are captured through iteratively training the CNNs using the FQI algorithm. Key findings of this study can be summarised as follows:

- Case studies demonstrate that the designed CNNs yield better learning accuracy compared to the DNNs, LSTM, and transformer networks when extracting key features from high-dimensional inputs. The CNNs save 68.40% and 78.07% of computational time from the LSTM and transformer networks, respectively. Electrification of heat and transportation causes doubled consumption in winter compared to summer.
- The proposed model can reduce 50% of the investment cost when the system transitions towards winter.
- The power loss and loss of load are managed within 5% compared to the benchmark optimisation.
- Our proposed model is scalable to various distribution networks in terms of the computational efficiency, by saving 68.18%, 82.89%, and 90.81% of computational time from the benchmark optimisation under the IEEE 18-bus, 33-bus, and 69-bus distribution networks, respectively.

From the perspective of industrial practices, the designed digital twins provide a transferable, scalable, and computational efficient model, for meeting the requirements of DSOs on exploiting physical data to assist their decision making.

Acknowledgement

This work was supported by the EPSRC through the project of "Analytical Middleware for Informed Distribution Networks (AMIDiNe)" (EP/S030131/1).

References

[1] Enabling the distribution system operation (dso)transition, Tech. rep., National Grid ESO (2021).
[2] M. Grieves, J. Vickers, Digital twin: Mitigating unpredictable, undesirable emergent behavior in complex systems, in: Transdisciplinary perspectives on complex systems, Springer, 2017, pp. 85–113.
[3] M. You, Q. Wang, H. Sun, I. Castro, J. Jiang, Digital twins based day-ahead integrated energy system scheduling under load and renewable energy uncertainties, Appl Energy 305 (2022) 117899.
[4] A. Saad, S. Faddel, T. Youssef, O. A. Mohammed, On the implementation of IoT-based digital twin for networked microgrids resiliency against cyber attacks, IEEE Trans Smart Grid 11 (6) (2020) 5138–5150. doi:10.1109/TSG.2020.3000958.
[5] J. Granacher, T.-V. Nguyen, R. Castro-Amoedo, F. Maréchal, Overcoming decision paralysis—a digital twin for decision making in energy system design, Appl Energy 306 (2022) 117954.
[6] E. O'Dwyer, I. Pan, R. Charlesworth, S. Butler, N. Shah, Integration of an energy management tool and digital twin for coordination and control of multi-vector smart energy systems, Sustain Cities Soc 62 (2020) 102412.
[7] Z. Huang, K. Soh, M. Islam, K. Chua, Digital twin driven life-cycle operation optimization for combined cooling heating and power-cold energy recovery (cchp-cer) system, Appl Energy 324 (2022) 119774.
[8] https://powerstar.com/ (Jan. 2022).
[9] https://www.ge.com/digital/ (Jan. 2022).
[10] https://new.siemens.com/global/en/products/energy/energy-automation-and-smart-grid/electrical-digital-twin.html (Jan. 2022).
[11] S. H. Oh, Y. T. Yoon, S. W. Kim, Online reconfiguration scheme of self-sufficient distribution network based on a reinforcement learning approach, Appl Energy 280 (2020) 115900.
[12] Y. Gao, W. Wang, J. Shi, N. Yu, Batch-constrained reinforcement learning for dynamic distribution network reconfiguration, IEEE Trans Smart Grid 11 (6) (2020) 5357–5369. doi:10.1109/TSG.2020.3005270.
[13] C. Lork, W.-T. Li, Y. Qin, Y. Zhou, C. Yuen, W. Tushar, T. K. Saha, An uncertainty-aware deep reinforcement learning framework for residential air conditioning energy management, Appl Energy 276 (2020) 115426.
[14] X. Zhang, W. Hua, Y. Liu, J. Duan, Z. Tang, J. Liu, Reinforcement learning for active distribution network planning based on monte carlo tree search, Int J Elec Power 138 (2022) 107885.

[15] Y. Zhou, A regression learner-based approach for battery cycling ageing prediction—advances in energy management strategy and techno-economic analysis, Energy 256 (2022) 124668.
[16] Y. Zhou, S. Zheng, Machine-learning based hybrid demand-side controller for high-rise office buildings with high energy flexibilities, Appl Energy 262 (2020) 114416.
[17] B. J. Claessens, P. Vrancx, F. Ruelens, Convolutional neural networks for automatic state-time feature extraction in reinforcement learning applied to residential load control, IEEE Trans Smart Grid 9 (4) (2016) 3259–3269.
[18] M. Kamruzzaman, N. Bhusal, M. Benidris, A convolutional neural network-based approach to composite power system reliability evaluation, Int J Elec Power 135 (2022) 107468.
[19] Z. Shi, W. Yao, L. Zeng, J. Wen, J. Fang, X. Ai, J. Wen, Convolutional neural network-based power system transient stability assessment and instability mode prediction, Appl Energy 263 (2020) 114586.
[20] L. Yin, J. Xie, Multi-temporal-spatial-scale temporal convolution network for short-term load forecasting of power systems, Appl Energy 283 (2021) 116328.
[21] R. Bellman, On the theory of dynamic programming, Proceedings of the National Academy of Sciences of the United States of America 38 (8) (1952) 716.
[22] `https://www.nationalgrideso.com/future-energy/future-energy-scenarios` (Dec. 2021).
[23] R. A. Davis, K.-S. Lii, D. N. Politis, Remarks on some nonparametric estimates of a density function, in: Selected Works of Murray Rosenblatt, Springer, 2011, pp. 95–100.
[24] M. D. McKay, R. J. Beckman, W. J. Conover, A comparison of three methods for selecting values of input variables in the analysis of output from a computer code, Technometrics 42 (1) (2000) 55–61.
[25] M. M. Deza, E. Deza, Encyclopedia of distances, in: Encyclopedia of distances, Springer, 2009, pp. 1–583.
[26] A. Ashouri, S. S. Fux, M. J. Benz, L. Guzzella, Optimal design and operation of building services using mixed-integer linear programming techniques, Energy 59 (2013) 365–376.
[27] H. Gao, L. Wang, J. Liu, Z. Wei, Integrated day-ahead scheduling considering active management in future smart distribution system, IEEE Trans Power Syst 33 (6) (2018) 6049–6061.
[28] S. H. Low, Convex relaxation of optimal power flow—part i: Formulations and equivalence, IEEE Trans Control Netw 1 (1) (2014) 15–27.
[29] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, et al., Pytorch: An imperative style, high-performance deep learning library, Advances in neural information processing systems 32 (2019).
[30] L. Thurner, A. Scheidler, F. Schäfer, J.-H. Menke, J. Dollichon, F. Meier, S. Meinecke, M. Braun, pandapower—an open-source python tool for convenient modeling, analysis, and optimization of electric power systems, IEEE Trans Power Syst 33 (6) (2018) 6510–6521.
[31] B. Bixby, The gurobi optimizer, Transp Re-search Part B 41 (2) (2007) 159–178.
[32] `https://www.amidine.net/` (Feb. 2023).
[33] `https://www.renewables.ninja/` (Mar. 2022).
[34] `https://pdf.sciencedirectassets.com/311593/1-s2.0-S2352340921X00030/1-s2.0-S2352340921003899` (Mar. 2022).
[35] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, arXiv preprint arXiv:1412.6980 (2014).
[36] C. Cortes, M. Mohri, A. Rostamizadeh, L2 regularization for learning kernels, arXiv preprint arXiv:1205.2653 (2012).
[37] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: a simple way to prevent neural networks from overfitting, J mach learn res 15 (1) (2014) 1929–1958.
[38] H. Pishro-Nik, Introduction to probability, statistics, and random processes (2016).
[39] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural computation 9 (8) (1997) 1735–1780.
[40] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, I. Polosukhin, Attention is all you need, Adv neural inf process syst 30 (2017).
[41] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, D. Wierstra, Continuous control with deep reinforcement learning, arXiv preprint arXiv:1509.02971 (2015).