Fall 12-16-2023

Quantum and Classical Learning Algorithms for Grid Integration of Energy Storage and Renewables: Operation, Modelling, and Planning

Bin Huang
Southern Methodist University, binhuangcn95@gmail.com

QUANTUM AND CLASSICAL LEARNING ALGORITHMS FOR GRID INTEGRATION OF ENERGY STORAGE AND RENEWABLES: OPERATION, MODELLING, AND PLANNING

Approved by:

Dr. Jianhui Wang
Professor
Electrical and Computer Engineering Department
Dissertation Committee Chairperson

Dr. Ping Gui
Professor
Electrical and Computer Engineering Department

Dr. Mohammad E. Khodayar
Associate Professor
Electrical and Computer Engineering Department

Dr. Barry Lee
Associate Professor
Mathematics Department

Dr. Miju Ahn
Assistant Professor
Operations Research and Engineering Management Department

Dr. Meng Yue
Electrical Engineer
Brookhaven National Laboratory

QUANTUM AND CLASSICAL LEARNING ALGORITHMS FOR GRID INTEGRATION OF ENERGY STORAGE AND RENEWABLES: OPERATION, MODELLING, AND PLANNING

A Dissertation Presented to the Graduate Faculty of the

Lyle School of Engineering

Southern Methodist University

in

Partial Fulfillment of the Requirements

for the degree of

Doctor of Philosophy

with a

Major in Electrical Engineering

by

Bin Huang

M.S., Electrical Engineering, South China University of Technology
B.S., Hydropower Engineering, Huazhong University of Science and Technology

December 16, 2023

ACKNOWLEDGMENTS

I would like to express my deepest appreciation and gratitude to my advisor, Prof. Jianhui Wang. Prof. Wang is not only exceptionally knowledgeable but also possesses a visionary outlook. Throughout my doctoral studies, he has provided me with invaluable guidance in my professional and personal endeavors, serving as a role model in my academic journey.

I would also like to extend my gratitude to all my supervisory committee members: Prof. Ping Gui, Prof. Mohammad E. Khodayar, Prof. Barry Lee, Prof. Miju Ahn, and Dr. Meng Yue. Additionally, I am deeply grateful to my mentors during my internship at the Interdisciplinary Science Department at Brookhaven National Laboratory, Dr. Meng Yue and Dr. Tianqiao Zhao.

Immense gratitude goes to lab colleagues for their invaluable academic discussions, brainstorming sessions, and daily companionship, including Dr. Xinan Wang, Dr. Shengfei Yin, Dr. Ying Zhang, Dr. You Lin, Dr. Xiaodong Zheng, Tao Wu, and Yanling Lin.

Words cannot express my profound gratitude for my father, mother, sister, girlfriend, and other family members. You are the most important people in my life and the pillars of my existence.

Lastly, I would like to extend gratitude to myself, a child from a small town in Guangdong, China. Never giving up on my dreams, I have always embraced life with passion and perseverance. Every step, through both the bitter and the sweet, has led me to this moment. It reminds me of Oscar Wilde's words, "We are all in the gutter, but some of us are looking at the stars."

Huang, Bin M.S., Electrical Engineering, South China University of Technology, 2019
B.S., Hydropower Engineering, Huazhong University of Science and Technology, 2016

Quantum and Classical Learning Algorithms for Grid Integration of Energy Storage and Renewables: Operation, Modelling, and Planning

Advisor: Dr. Jianhui Wang

Doctor of Philosophy conferred December 16, 2023

Dissertation completed October 16, 2023

With the pressing net-zero objectives, renewable energy sources (RESs) and energy storage systems (ESSs), which serve as the main force in providing low-carbon energy and flexibility, are being integrated into the power systems at an unprecedented rate. The rising penetration trend of RESs and ESSs brings both opportunities and challenges. In the short term, the inherent intermittency of RESs and high capital costs of ESSs introduce obstacles to revenue maximization for hybrid power plants and complicate the provision of privacy-ensured, high-fidelity network modeling. Long-term planning faces the hurdle of expanding ESS capacity in microgrids while maintaining power supply resilience with specific frequency characteristics. The recent surge in machine learning and quantum computing offers potential solutions to address these key challenges. Aiming to achieve secure and economic operation through capacity scheduling of investor-owned photovoltaic-battery storage systems (PV-BSS), this dissertation utilizes a Proximal Policy Optimization (PPO)-based deep reinforcement learning (DRL) agent that can handle continuous action spaces and ensure safety constraints, addressing challenges in adapting to volatile market signals and PV generation profiles. To enhance the accuracy of equivalent models for active distribution networks (EMADNs) in the presence of high RES penetration and associated management techniques, an adaptive EMADN with tunable scales and parameters is developed, featuring a leaves-trimming topological reduction method and a distributed PPO-based agent. Then, this dissertation seeks to strategize BSS expansion planning for microgrids by introducing a two-stage multi-period framework that combines the quantile regression DRL algorithm with linear programming, ensuring adaptive planning for long-term variations in RES, load, and battery pricing. Finally, this dissertation aims to optimize energy arbitrage tasks in ESSs by introducing a quantum policy learning algorithm that involves a strategically structured variational quantum circuit, offering a novel approach to action exploration and maintaining operational safety within energy markets. Case studies based on real electricity market data and RES profiles validate the effectiveness and benefits of the proposed methodologies compared to state-of-the-art techniques.

TABLE OF CONTENTS

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

Chapter 1

Introduction

1.1 Background

The net-zero goal induced by global warming is bringing revolutionary transformations to the power industry. As a major means to fulfill decarbonization, renewable energy sources (RESs) are experiencing exponential growth. U.S. Energy Information Administration (EIA) points out in its Annual Energy Outlook 2022 that power generation from RESs in the United States will jump from 21% (about 0.84TkWh) in 2021 to 44% (about 2.42TkWh) in 2050 [1]. Among RESs, solar energy and wind power have become the center of attention, accounting for 51% and 31% of RES generation, respectively, in 2050.

Despite the promising low-carbon vision provided by RES, RES has its inherent limitations, among which stochasticity and intermittency are the most significant. Energy storage is an essential technical way to resolve RES issues. Due to the flexibility and agility of charging and discharging, energy storage can readily solve the unexpected power shortage or surplus of RES generation. Among numerous energy storage technologies, battery energy storage (BES) is currently the most economically viable utility-scale energy storage solution [2]. Moreover, the economic feasibility brought by the development of BES has increased its competitiveness. For instance, National Renewable Energy Laboratory projects a 30% to 70% cost plummet for 4-hour lithium-ion systems relative to 2020 [3]. Therefore, BES and its storage peers are indispensable components of modern power grids.

Grid connections of inverter-interfaced RES and energy storage components bring more variability to the grid, creating the need for more powerful information and communications architecture, in which measurement and monitoring instruments are primary parts [4,5]. At the distribution and microgrid (MG) level, the deployment of micro phasor measurement units (μPMUs) and smart meters (SMs) are proliferating. μPMUs is a high-speed and low-cost synchrophasor that captures and streams voltage and current measurements at 30 frames per second, or more [6]. The time stamps associated with Global Positioning System are utilized to attain time synchronization. The initiative of advanced metering infrastructure (AMI) stimulates the installation of SMs, enabling two-way communication of instantaneous electricity usage and control signals between consumers and suppliers. EIA reveals that more than 111 million smart meters were deployed in the US as of the end of 2021 [7]. The installation of smart meters reached 665 million units globally in 2017. According to a recent report, the global smart-meter market will be valued at USD 19.7 billion in 2021. By the end of 2028, it is projected to generate revenues of about 34.5 billion USD. Meteorological instrumentation for renewable energy availability forecasting is also increasingly used, resulting in the availability of real-time meteorological data such as solar radiation, wind speed, temperature, and precipitation [8]. The availability of multi-mode and multi-time-scale data in power grids accelerates the arrival of the big data era, bringing both opportunities and challenges [9–12].

The recent advances in machine learning (ML) and deep learning (DL) techniques are the cruces to potential opportunities and challenges in the big data era [13–16]. The deeper architectures result in better representation learning and data-possessing capability. The phenomenal performances of deep neural networks (DNNs) facilitate the development of data-driven solutions to those physics-related problems in which the physical mechanism is not fully understood or in which it is intractable to derive a high-resolution numerical solution with limited computational resources [17–19]. The breakthroughs in DL gave rise to deep reinforcement learning (DRL), which uses DNNs to approximate the value functions, the policies, and even the environment itself. DRL is a trial-and-error approach in which the decision-making agent takes actions based on observed states and updates its policies based on rewards received from the environment with the evolvement of value functions. The disruptive achievements of the DRL in playing game [20], protein structure prediction [21], numerical algorithm development [22], etc., attract the gaze from all over the world. The capabilities of the DRL on autonomous learning without data labeling, fast execution efficiency, and generalization to unseen scenarios are commendable. However, it is not straightaway to transplant and apply DL and DRL to the power grids. The challenges can be summarized as follows. First, the training of the DNNs often requires a great quantity of high-quality training data to guarantee performance. However, compared with images, texts, and game simulators, training data may be very scarce or costly to obtain in power grid applications. Second, the uses of the DNN are also plagued by the black-box nature of DNNs and the lack of theoretical analysis of the lower bound of performance, resulting in doubts about the reliability of the results, especially in safety-critical applications [23–26]. Third, the plain DNNs are likely to produce physically infeasible results without a delicate design and domain-specific knowledge from power systems [27, 28].

Besides DL and ML, the recent breakthroughs in quantum computing (QC) by tech giants like Google have also captured significant attention. Google achieved a milestone in QC by demonstrating that a specific quantum processor could perform a task in mere minutes, which would take a state-of-the-art classical supercomputer thousands of years. This experiment showcased the potential of quantum systems to outperform classical counterparts in certain computational tasks, thereby illustrating quantum supremacy [29]. On the other hand, the development of quantum algorithms, such as Shor's algorithm for factorization [30] and Grover's for database searching [31], has underscored the theoretical superiority of quantum computing over classical paradigms. Quantum machine learning algorithms, centered around variational quantum circuits, as a fusion of QC and ML, are emerging. Moreover, given that QML is a nascent field, its applications in the energy system domain are still sparse. Considering the current noisy intermediate-scale quantum (NISQ) era, the inherent noise in quantum computers remains significant. As such, the practical implementation of QML in power systems necessitates further exploration and a deeper evaluation.

1.2 Objective and Scope of Research

This thesis aims to develop grid-informed learning algorithms for integrating large-scale renewable energy into power grids and unravel the power of DL, DRL, and QC in building a resilient, economical, and environmentally friendly power grid. The subsequent four technical chapters investigate and discuss the operation, modeling, and planning problems of modern power systems [32–34]. Table 1.1 summarizes the technical details of these three chapters. The objects of study involve PV-battery, distribution network, and MG. The time

Table 1.1: Summary of four technical chapters.

Chapter	Problem	Object	Time-scale	Time resolution	Source of uncertainty	Grid interaction
2	capacity scheduling	PV-battery	weekly	hourly, every 4s	market signal, PV	market participation
3	model reduction	distribution network	weekly	every 15mins	PV, wind, load	load flow
4	storage planning	microgrid	25 years	five-year, hourly	storage price, PV, wind, load	optimal power flow
5	energy arbitrage	energy storage	weekly	hourly	price signal	energy market

span ranges from seconds to years. Uncertainties from different sources, including market, weather, and technical developments, are fully considered. Depending on the applications, the angles of the combination of the learning algorithms and the grid are various. By delineating the underlying physical and market mechanisms, this thesis strives to provide reliable QC/DRL-aided solutions to the grids.

The following subsections outline the objective and scope of research in Chapters 2, 3, 4, and 5, in sequence.

1.2.1 Deep Reinforcement Learning-based Capacity Scheduling for PV-Battery Storage System

Investor-owned photovoltaic-battery storage systems (PV-BSS) can gain revenue by providing stacked services, including PV charging and frequency regulation, and by performing energy arbitrage. Capacity scheduling (CS) is a crucial component of PV-BSS energy management, aiming to ensure the secure and economic operation of the PV-BSS. This work proposes a Proximal Policy Optimization (PPO)-based deep reinforcement learning agent to perform the CS of PV-BSS. Unlike previous work that uses value-based methods with the discrete action space, PPO can readily handle continuous action space and determine the specific amount of charging/discharging. To enforce the safety constraints of BSS's energy and power capacity, a safety control algorithm using a serial strategy is proposed to cooperate with the PPO agent. The PPO agent can exploit the capacity of BSS safely while maximizing the accumulated net revenue. After training, the PPO agent can adapt to the highly uncertain and volatile market signals and PV generation profiles. The efficacy of the proposed CS scheme is substantiated by using real market data. The comparative results demonstrate that the PPO agent outperforms the Deep Deterministic Policy Gradient agent, Advantage Actor-Critic agent, and Double Deep Q Network agent in terms of profitability and sample efficiency.

1.2.2 Adaptive Static Equivalences for Active Distribution Networks with Massive Low-Carbon Energy Integration: A Distributed Deep Reinforcement Learning Approach

Active distribution networks (ADNs) with 100% renewable energy sources penetration are one of the promising segments in the net-zeros emission power sector. As an efficient analysis tool, the equivalent model for ADNs (EMADNs) can be leveraged to expedite the lengthy analysis and preserve data privacy. Nevertheless, volatile RESs, along with their active management techniques, e.g., voltage regulation schemes (VRSs) of RESs, deteriorate the fidelity of the developed EMADNs derived from historical data. This work proposes an adaptive EMADN, featuring tunable network scales and adaptive parameters to address the above challenges. A leaves-trimming network topological reduction algorithm with a customized reduction degree is utilized to preserve the radial topology, the nodes of interest, and VRSs. A distributed Proximal Policy Optimization (DPPO)-based deep reinforcement learning agent is proposed to adjust the parameters periodically to maintain the fidelity of EMADNs with massive RESs. The distributed training scheme of the DPPO exploits the multi-processors to boost training efficiency. Case studies on the IEEE-33 bus and IEEE 123-bus ADNs with massive photovoltaic and wind generation demonstrate the superior accuracy and efficiency of the proposed agent-based EMADNs in various scenarios, especially when the production of RESs exceeds the load amount.

1.2.3 Two-Stage Frequency-Constrained Adaptive Storage Expansion Strategy for Microgrids Using Deep Reinforcement Learning

Battery energy storage (BES) is a versatile resource for the secure and economic operation of microgrids (MGs). Prevailing stochastic optimization-based approaches for BES expansion planning for MGs are computationally complicated. This work proposes a data-driven bi-level multi-period BES expansion planning framework to determine the siting, sizing, and timing of BES installations. The proposed planning framework unifies deep reinforcement learning (DRL) and linear programming, thereby decoupling the determinations for the integer and continuous decision variables in two time scales, respectively. In the upper level, a rainbow DRL agent with quantile regression is trained to provide dynamic planning policies to accommodate stochastic renewable energy resources (RESs), load, and battery price changes efficiently. The lower level computes the optimal operation of MGs with frequency constraints to hedge the islanding contingency. The two levels communicate with one another by exchanging storage configuration and operating expenses in order to accomplish the shared goal of minimizing investment and operation costs. Comparative case studies on an MG are carried out to demonstrate the superiority of the proposed DRL-based solution to the mixed-integer linear programming counterpart on efficiency, scalability, and adaptability.

1.2.4 Quantum Policy Learning for Energy Storage Arbitrage

The advent of quantum computing paves a new pathway for providing a more efficient and data-driven solution with enhanced learning capabilities for energy arbitrage (EA) tasks in energy storage systems (ESSs). This work proposes a Quantum Policy Learning (QPL) algorithm to streamline the online EA processes of ESSs within the energy markets. A strategically constructed Markov Decision Process is utilized to encourage early-stage proactive exploration of the action space while ensuring the operational safety of ESSs. In contrast to the traditional reinforcement learning paradigm, which employs neural networks to approximate the policy, QPL leverages a Variational Quantum Circuit (VQC) as the parameterized policy representation. Given the noisy quantum hardware prevalent in the Noisy Intermediate-Scale Quantum era, the VQC in QPL adopts a design characterized by low depth and width, composed of intricately designed alternating variational entanglement sub-layers and variational embedding sub-layers. The loss function and gradient calculations for training the parameters of the quantum policy are derived, incorporating considerations for action re-parameterization techniques, squashing, and linear transformations. In case studies, the proposed QPL method highlights a quantum advantage attributed to the superior expressiveness of the devised VQC, enabling more rapid convergence and optimization performance comparable to classical algorithms yet requiring merely 1.6% of the parameter quantity (i.e., only 20 parameters needed). Subsequent experiments conducted on the IBM_Lagos quantum hardware quantified the impact of gate errors and qubit decoherence on the online operation of QPL, corroborating the noise resistance of the QPL approach. A series of comparative experiments are also employed to scrutinize the optimization performance, training stability, and compilation efficiency of the QPL.

Chapter 2

Deep Reinforcement Learning-based Capacity Scheduling for PV-Battery Storage System

This section addresses the capacity scheduling for investor-owned photovoltaic-battery storage systems to provide multiple services including energy arbitrage, requency regulation, and PV charging. Leveraging a Proximal Policy Optimization-based deep reinforcement learning approach, which surpasses traditional value-based methods, the proposed model effectively manages charging/discharging in response to uncertain market signals and photovoltaic profiles, demonstrating superior profitability and efficiency compared to other state-of-the-art agents in real-market scenarios.

2.1 Introduction

The appeal for the low-carbon future spurs the increasing integration of renewable electricity generation, including utility-scale photovoltaic (PV) systems, to the power grid. This trend also brings significant challenges to the stability and reliability [35] of the operation of the power grid due to the limited predictability and controllability of renewable sources. Through providing great flexibility and smoothing power fluctuation, battery storage systems (BSSs) are proven to be an effective solution to the extensive integration of PV. The decrease of the capital cost of BSSs facilitates the development of the emerging co-located PV-BSS [36–38], which consists of one or multiple PV plants and BSSs. The trend of combining PV energy with battery storage makes PV generation increasingly competitive.

Investor-owned PV-BSS can be regarded as an independent entity to the power grid, with the goal of maximizing the revenue. Developing optimal scheduling strategies for the PV-BSS has a huge influence on the revenue of the existing systems and on the economic appraisal of the potential PV-BSS projects, spurring a substantial body of research. Due to the prominent flexibility and fast-response feature, BSSs can provide multiple services associated with multiple revenue streams, including peak shaving [39], reserve [40], energy arbitrage (EA) [41], frequency regulation (FR) [37], etc. It is reported in [39, 40] that by providing the stacked services, the owners of the BSS can make full use of the battery and earn extra profit.

The conventional approaches to address the capacity scheduling (CS) problem of BSSs, which provides stacked services are stochastic programming (SP) approaches [40], robust optimization (RO) approaches [42], and model predictive control (MPC) [43]. Ref. [40] presents an optimal joint bidding strategy of BSS in the day-ahead market using scenario reduction techniques. There is a tradeoff between the model granularity and computation efficiency in [40]. The BSS bidding problem in [42] is solved via iterating through a master problem and an availability check max-min subproblem. Karush-Kuhn-Tucker (KKT) conditions are used to transform the subproblem, which is solved by column & constraint generation eventually. A stochastic MPC framework [43] is introduced to determine the commitments of BSS in energy and FR markets on both the real-time and long-term time scales. However, the bidding of frequency regulation capacity is not accounted for. Though the optimization-based approaches have been making significant advances, applying the solution of such approaches to the real-world is limited because this kind of approach is dependent on the assumption on the prior distribution of the random variables. For example, the assumptions upon the distribution or the range of the random variables and the convexity of the optimization problem are indispensable in most cases [40, 42–44]. It is reported in [45] that it is still challenging to solve the optimal battery control problem or give a guarantee on any theoretical performance without a strong assumption of the random signals. In most cases, only the historical data of the random variables rather than the predefined distributions are available, and it is tricky to formulate the problem as a convex optimization problem. Besides, the SP approach in [40] suffers from computational intractability when it encounters the highly uncertain environment and relatively long scheduling cycle.

Recently, leveraging the advancement of deep learning and reinforcement learning (RL), deep reinforcement learning (DRL) has aroused great interest in the academia and industry [20]. In the field of smart grid, researchers have utilized DRL to address numerous knotty problems, e.g., autonomous voltage control [46], autonomous multi-energy management [47], electric vehicle charging scheduling [48].

The fully data-driven DRL algorithms are the ideal approaches to tackle the CS problem with strong uncertainties and long scheduling cycles. First and foremost, considering the random nature of the PV generation and market signals, and the time-coupled feature of the state of charge (SOC), the CS of PV-BSS is essentially a discrete stochastic control process, which can be modelled as Markov Decision Process (MDP). DRL agents are notable for addressing such a problem. In contrast to the SP-type methods dependent on the probability density functions (PDFs) of random variables, DRL optimizes policy directly on the basis of the historical/simulation data. DRL algorithms outperform traditional optimization techniques in terms of adaptivity. Different from the conventional optimization techniques, which requires reformulation and recalculation for various environments, DRL can output the policy that is applicable to volatile and various environments. What's more, once trained, DRL agents can provide decent scheduling results on test data, i.e. data that is not accessible during the training phase, without the need to reformulate and retrain. This phenomenal adaptivity is partially attributed to the powerful function approximation function of the neural network.

Q-learning and Double Deep Q Networks (DDQN) have been used in [41,49,50] to control the charging/discharging of batteries. However, since [41, 49, 50] focus on single service only, namely EA, they all discretize the action space of the battery. For example, in [49, 50], the statuses of batteries, which consists of charge, idle, and discharge, are determined by the DRL agent, neglecting the specific decision on the amount. The action space of [41] is discretized into five parts, which include the maximum and half maximum charge/discharge power capacity, and zero. Though significant progress has been made in [41,49,50], the assumption of the discrete action space does not hold in the context of conducting CS of batteries between stacked services. The precise and specific amount of the charging/discharging power capacity should be determined to fully exploit the profitability of stacked services, which necessitates the adoption of continuous action spaces.

This work employs Proximal Policy Optimization (PPO)-based DRL to dispatch the capacity of PV-BSS. PPO is a cutting-edge DRL algorithm developed in [51], which is the variant of Trust Region Policy Optimization (TRPO) [52] and Advantage Actor-Critic (A2C) [53]. Similar to TRPO, PPO can guarantee the safe exploration of the agent by scrutinizing the distance between the updated policy and the previous policy. PPO can be implemented in a more efficient manner by avoiding tackling the complicated second-order optimization problem in TRPO. More importantly, PPO can tackle the continuous and multi-dimensional action space readily, which is able to exploit the potential of the BSS providing stacked services.

The contributions of this work are summarized as follows:

1. A PPO-based DRL approach to learning the safe and optimized CS policy for the PV-BSS in the context of performing the stacked services is proposed in Section II-A. Two essential charters in the DRL algorithm, i.e. the environment and the DRL, are specified as a safety control algorithm and a PPO agent, respectively.

2. A safety control algorithm (SCA) for PV-BSS is proposed in Section 2.2.3, which can coordinate the scheduling of multiple services of the PV-BSS, including frequency regulation, PV charging, and energy arbitrage. In the proposed algorithm, the time-coupled characteristics of SOC, the safe operation constraints of SOC, and the upward/downward constraints of power capacity are strictly satisfied. SCA features the serial decision-making process, which eliminates the inclusion of the penalties of the constraint violation on the reward function for the DRL agent and thus avoids the heuristic design of the penalty coefficients.

3. A PPO agent, which serves as the energy management unit by perceiving the system status and releasing the control signal, is developed in Section III-B. Unlike tradition optimization techniques, the fully data-driven DRL agents are being trained upon the volatile training data directly and can adapt well to the volatile and various environments characterized by the significant uncertainties from PV generation, price, and market signals. The PPO agent features the adoption of a clipped surrogate objective function, which is beneficial to the sample efficiency and the convergence rate. Besides, in contrast to most value function-based DRL agents which are only applicable to the discrete action space, the PPO agent is characterized by the continuous action space, resulting in the better exploitation of the profitability of the stacked services.

4. Case studies are carried out using real market data, which are shown in Section IV. Through the comparisons with the Deep Deterministic Policy Gradient (DDPG), A2C, and DDQN agents, the practicability and superiority of the PPO agent are corroborated.

The rest of the work is organized as follows. Section II proposes a safety control algorithm for PV-BSS to perform stacked services. Section III describes the proposed PPO-based DRL approach and the training scheme of the PPO agent. Case studies are conducted in Section IV using the real data from the PJM energy and regulation market, which demonstrates the practicality and superiority of the proposed scheduling scheme. Finally, concluding remarks are given in Section V.

2.2 Safety Control Algorithm of PV-BSS to Perform Stacked Services

In this section, an overview of the PV-BSS is introduced first, in which the system components and functions are presented. Then, the models of stacked services and their individual revenue models are detailed, followed by analyzing the profitability and the impact of these stacked services on the battery power and energy capacity. Based on the analysis, a safety control algorithm of PV-BSS to perform stacked services is proposed to ensure the safe operation of the PV-BSS while maximizing the system benefits.

2.2.1 System Overview: Components and Functions

Fig. 2.1 shows a schematic diagram of the investor-owned PV-BSS, which consists of three core components, i.e., PV generation system, BSS, and energy management unit (EMU). The PV generation system utilizes the solar panel to transform solar energy into electricity, which can be stored into the battery or be sold to the power grid directly through the PV inverter. As an intermediate, BSS interacts closely with the power grid and PV generation systems via charging/discharging while EMU functions as the core of the PV-BSS. It collects the predicted PV power, the state of BSS, and market signals such as energy and regulation market prices [54]. With the collected information, EMU dispatches the available battery capacity to maximize the long-term cumulative revenue while guaranteeing the battery's secure operation. There are two types of lines in the diagram; one is the dotted lines that represent the flow of information in the system. The flow of information is made up of the prerequisite input data needed for the EMU to make decisions and the output data that represents the results of the EMU's decisions. The other is thick solid lines representing the physical energy flow of the system.

In our work, with the superiority of addressing sequential decision-making problems, the DRL agent is leveraged to perform the energy management task. The main characters in the DRL algorithm are the DRL agent and the environment. The general goal of the DRL agents is learning a policy to maximize the expected utility via the trial and error interaction with the environment.

For the specific task in this work, the agent-environment interaction loop for the DRL algorithm is plotted in Fig. 2.2. The environment is the world that the agent lives in and interacts with. At every step of interaction, the agent sees a (possibly partial) observation of the state of the world and then decides on an action to take. The environment changes when the agent acts on it, but may also change on its own. The safety control algorithm, which is elaborated in Section II.C, can be interpreted as the environment to the DRL agent. As shown in Fig.2.2, the state is set as a synthesis of available PV power, battery status, and market signals, which are the input information flow into EMU in Fig. 2.1 as well. The action is set as control signals for the PV-BSS, which are the output control signals from EMU in Fig. 2.1. α, β, and ξ are the ratio coefficients for the stacked services, which will be elaborated in Section II-C. The agent also perceives a reward signal from the environment, a number that tells it how good or bad the current state is. In this case, the net revenue from providing stacked services is used as the reward signal. Further explanations and mathematical formulation for the state, action, and reward are presented in Sec.2.3.1. The detailed formulation and implementation of the DRL algorithm are presented in Sec.2.3.2.

2.2.2 Stacked Services

For the PV-BSS, three primary services are taken into account, namely, PV charging, FR, and EA. Although providing multiple services can bring more economic benefits, it also brings challenges to battery control. One of the significant challenges is that storage capacity is shared between the stacked services dynamically. In other words, multiple services share the same dispatchable capacity simultaneously. Furthermore, the charging/discharging

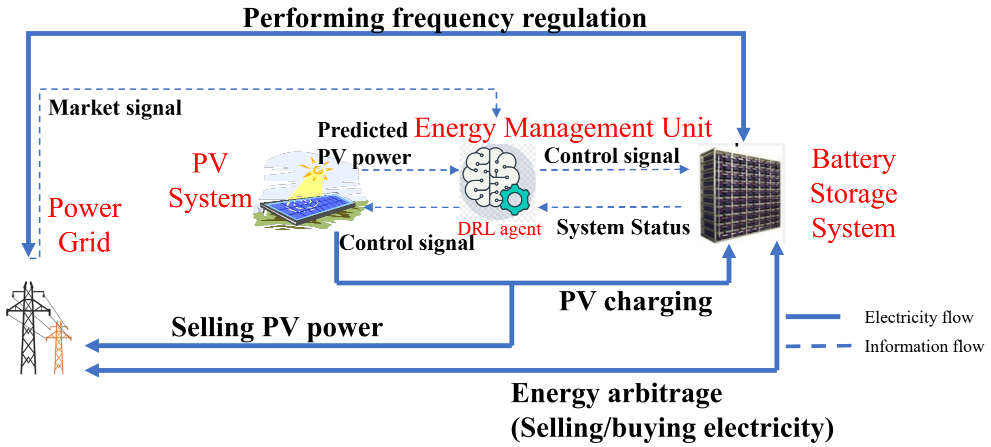


Figure 2.1: Schematic diagram of the PV-BSS.

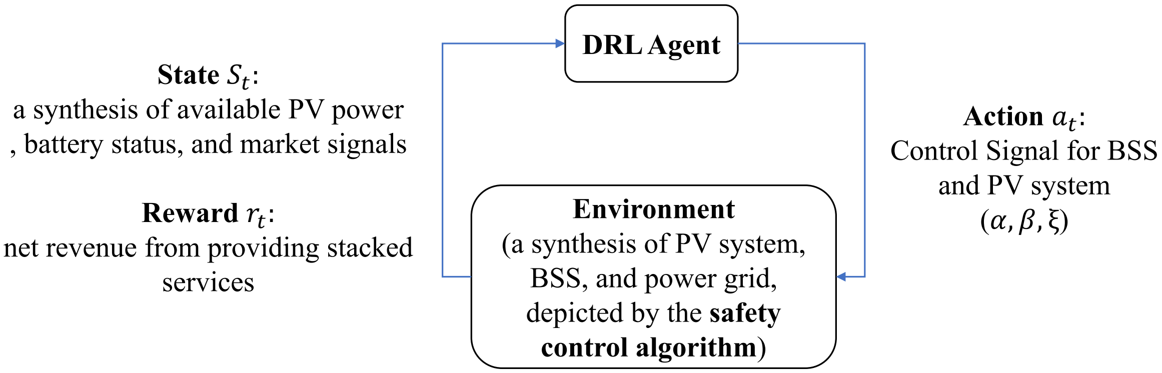


Figure 2.2: Agent-environment interaction loop.

behaviour is constrained by the physical capability of the battery, i.e., power and energy capacity limits. EMU should coordinate these services and exploit the charging/discharging capability of BSS to the full extent.

2.2.2.1 Fast Frequency Regulation Service

PJM has a relatively mature regulation market, so this work focuses on the market mechanism of PJM, in which most BSSs are committed to FR by tracking the Regulation D (RegD) signal. It is noted that the control algorithm proposed in this work can be generalized to other markets. It is assumed that the role of PV-BSS in the FR market is a price-taker, which means that it must accept the prevailing prices in the market.

In this work, the scheduling of BSSs is on an hourly basis. Although the RegD signal is designed with the feature of approximate energy neutrality, a battery still has a hourly fractional energy loss [55, 56]:

$$q_t = \sum_{j=1}^{J} \left(\frac{\delta_{j,t}^{+}}{\eta_{\text{dis}}} + \delta_{j,t}^{-} \cdot \eta_{\text{ch}} \right) \cdot \Delta t \tag{2.1}$$

where hour t is divided into J time intervals; $\delta_{j,t}^{+/-}$ ($|\delta_{j,t}^{+/-}| \leq 1$) is the j-th RegD signal at hour t. "+" and "-" denote the regulation-up (discharging) signal and regulation-down (charging) signal, respectively. η_{ch} and η_{dis} are the battery charge/discharge efficiencies, respectively. Δt represents the time interval of RegD signals, and it is set to be 4s in PJM. It is noted that the subscript t is used as the index for hour t throughout the work. The positive/negative q_t indicates that the BSS will discharge/charge, respectively, through the provision of FR.

BSS will be reimbursed dependent on the deployed regulation power capacity F_t^{f} [40, 55]:

$$B_t^{\text{f}} = P_t^{\text{f}} \cdot \varphi_t \cdot \left(\lambda_t F_t^{\text{PCP}} + F_t^{\text{CCP}} \right) \tag{2.2}$$

where B_t^{f} is the BSS's revenue on providing FR; λ_t is the mileage ratio; φ_t is the performance score; F_t^{PCP} and F_t^{CCP} are the performance/capacity clearing prices, respectively.

2.2.2.2 PV Charging

The available PV generation can be either sold directly in the energy market or stored by BSS to perform the EA and FR in the future.

Denote $\overline{P}_t^{\mathrm{pv}}$ as the available solar power, which can be divided into two parts, i.e., charging power $P_t^{\mathrm{pv,e}}$ and selling power $P_t^{\mathrm{pv,s}}$. The revenue of selling PV power can be calculated as:

$$B_t^{\mathrm{pv}} = P_t^{\mathrm{pv,s}} \cdot \Delta h \cdot F_t^{\mathrm{lmp}} \tag{2.3}$$

where Δh is the time duration and is set to be 1 hour in this work; F_t^{lmp} is the locational marginal price (LMP) of the energy market.

2.2.2.3 Energy Arbitrage

EA is a measure adopted by the operators of the BSS to take advantage of the price differential between hours. Denote P_t^{EA} as the power capacity deployed for EA. Positive and negative P_t^{EA} denote buying and selling electricity, respectively. The remuneration of performing EA is calculated as:

$$B_t^{\mathrm{EA}} = -P_t^{\mathrm{EA}} \cdot \Delta h \cdot F_t^{\mathrm{lmp}} \tag{2.4}$$

2.2.2.4 Sequence of Stacked Services

To take into account the physical capacity characteristics of the battery and to avoid the DRL agent making decisions that violate capacity constraints, scheduling can be made in the form of a proportional factor based on BSS's available power capacity. Unlike the solution from the mathematical optimization models that can be applied to the CS of battery in parallel, CS by DRL is arranged in a serial strategy based on service type in this work. In a serial strategy, whenever a service is arranged, the available power/energy capacity is updated. Fig.2.3 shows an example for differentiating serial and parallel strategies. Assume the blank rectangle represents the BSS's available power capacity. The parallel strategy dispatches the capacity for each service simultaneously. However, the serial strategy dispatches the capacity for each service in sequence. Whenever a service is arranged, the available power/energy capacity is updated. The motivation of the serial strategy is to boost the convergence of the DRL agent.

The sequence of services does not impact the optimality of the DRL agent, as DRL has the ability to seek long-term cumulative gains in a changing environment. However, the sequence of services will affect how quickly the algorithm converges, which is discussed further in the case studies. In the following statement, to make our statement clearer, we assume that CS is based on the following priorities: the proportion factor for FR is determined first, then PV charging, and finally, EA.

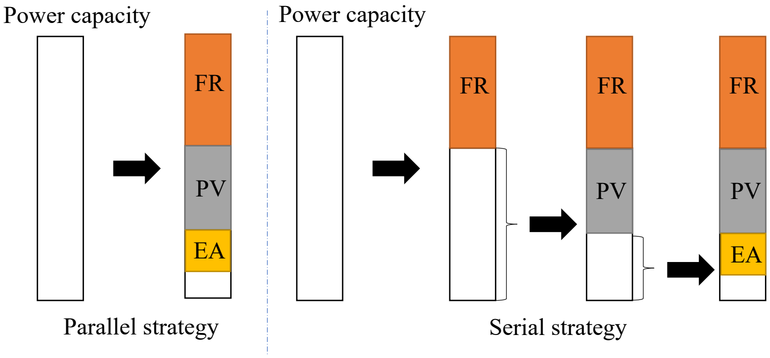


Figure 2.3: Parallel and serial strategy.

2.2.3 Safety Control Algorithm

The safety control algorithm can ensure that the operating constraints of the battery are strictly satisfied, and it is detailed in Algorithm 1, where $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, and $\boldsymbol{\xi}$ are the PV charging, EA, and FR ratio coefficients, respectively. They represent the control policy of the EMU, and they are generated by the DRL agent simultaneously. The bound for $\boldsymbol{\alpha}$ and $\boldsymbol{\xi}$ is $[0, 1]$, whereas the bound for $\boldsymbol{\beta}$ is $[-1, 1]$.

For hour t, based on SOC_t, the upward and downward feasible region of SOC can be derived, respectively:

$$\begin{aligned} SOC_t^{\text{up}} &= \overline{SOC} - SOC_t \\ SOC_t^{\text{dn}} &= SOC_t - \underline{SOC} \end{aligned} \tag{2.5}$$

Denote $P_{\max}^{\mathrm{up}}$ and $P_{\max}^{\mathrm{dn}}$ as maximum upward and downward power capacities of BSS, respectively. It is noted that $P_{\max}^{\mathrm{up}} > 0$ and $P_{\max}^{\mathrm{dn}} < 0$ in the notation of this work.

Among three services, the PV-BSS first dispatches the capacity for FR. To ensure the charging power will not cause the violation of the upper limit of SOC, the maximum possible FR ratio coefficient is derived:

$$\xi_{\max} = \begin{cases} \dfrac{SOC_t^{\mathrm{up}} U}{-P_{\max}^{\mathrm{up}} \cdot q_t}, & \text{for } q_t \leq 0 \qquad (2.6) \\ \dfrac{SOC_t^{\mathrm{dn}} U}{-P_{\max}^{\mathrm{dn}} \cdot q_t}, & \text{for } q_t > 0 \qquad (2.7) \end{cases}$$

where the product of $P_{\max}^{\mathrm{up}}/P_{\max}^{\mathrm{dn}}$ and q_t is the maximum possible energy gain for a battery from FR at hour t; U is the energy capacity.

After the $\xi_{\max}$ is available, we can impose the clip function clip on ξ_t, which is the raw control signal from the DRL agent, to prevent the over-charging. The clip function can clamp all elements in input into the range [min, max]. Let min_value and max_value be min and max, respectively, clip function returns:

$$y_i = \min(\max(x_i, \mathrm{min\_value}), \mathrm{max\_value}) \tag{2.8}$$

In this case, the min_value and max_value are set to 0 and $\xi_{\max}$, respectively.

Then P_t^{f} is determined by:

$$P_t^{\mathrm{f}} = \begin{cases} \xi_t P_{\max}^{\mathrm{up}}, & \text{for } q_t \leq 0 \qquad (2.9) \\ -\xi_t P_{\max}^{\mathrm{dn}}, & \text{for } q_t > 0 \qquad (2.10) \end{cases}$$

One exception is that if P_t^{f} is less than the minimum bidding capacity specified in the frequency regulation market, BSS will not be able to participate in the regulation market at hour t.

Due to the charging/discharging by FR, the upward/downward space of SOC is updated as:

$$SOC_t^{\mathrm{up}} \leftarrow SOC_t^{\mathrm{up}} + \frac{P_t^{\mathrm{f}} q_t}{U}, \text{for } q_t \leq 0 \tag{2.11}$$

$$SOC_t^{\mathrm{dn}} \leftarrow SOC_t^{\mathrm{dn}} - \frac{P_t^{\mathrm{f}} q_t}{U}, \text{for } q_t > 0 \tag{2.12}$$

Similarly, the remaining maximum upward/downward power capacity of BSS at time t can be obtained:

$$P_{\max}^{\mathrm{up}} \leftarrow P_{\max}^{\mathrm{up},0} - P_t^{\mathrm{f}}, \text{for } q_t \leq 0 \tag{2.13}$$

$$P_{\max}^{\mathrm{dn}} \leftarrow P_{\max}^{\mathrm{dn},0} + P_t^{\mathrm{f}}, \text{for } q_t > 0 \tag{2.14}$$

If the current PV generation is available $\overline{P}_t^{\mathrm{pv}} > 0$, derive the maximum possible PV charging ratio coefficient to prevent the overcharging:

$$\alpha_{\max} = \frac{SOC_t^{\mathrm{up}} U}{\overline{P}_t^{\mathrm{pv}} \Delta h \cdot \eta_{\mathrm{ch}}} \tag{2.15}$$

Then impose the clip function on α_t to hamper the excessive charging of the BSS. Besides, the power capacity dispatched for the PV charging should be limited as follows:

$$P_t^{\mathrm{pv,e}} = \mathrm{clip}(\overline{P}_t^{\mathrm{pv}} \cdot \alpha_t, 0, P_{\max}^{\mathrm{up}}) \tag{2.16}$$

Due to the PV charging, the upward space of SOC is updated as:

$$SOC_t^{\mathrm{up}} \leftarrow SOC_t^{\mathrm{up}} - \frac{P_t^{\mathrm{pv,e}} \Delta h \cdot \eta_{\mathrm{ch}}}{U} \tag{2.17}$$

Similarly, $P_{\max}^{\mathrm{up}}$ is updated again as:

$$P_{\max}^{\mathrm{up}} \leftarrow P_{\max}^{\mathrm{up}} - P_t^{\mathrm{pv,e}} \tag{2.18}$$

The direction of EA is indicated by the sign of β_t. Positive and negative β_t indicate BSS purchasing and selling electricity, respectively. The maximum possible EA ratio coefficient is derived as:

$$\beta_{\max} = \begin{cases} \dfrac{SOC_t^{\text{up}} U}{P_{\max}^{\text{up}} \Delta h \cdot \eta_{\text{ch}}}, & \text{for } \beta_t \geq 0 \qquad (2.19) \\ \dfrac{SOC_t^{\text{dn}} U}{-P_{\max}^{\text{dn}} \cdot \Delta h / \eta_{\text{dis}}}, & \text{for } \beta_t < 0 \qquad (2.20) \end{cases}$$

After clipping β_t with:

$$\beta_t = \text{clip}(\beta, 0, \beta_{\max}), \tag{2.21}$$

P_t^{EA} is calculated as:

$$P_t^{\text{EA}} = \begin{cases} P_{\max}^{\text{up}} \cdot \beta_t, & \text{for } \beta_t \geq 0 \qquad (2.22) \\ -P_{\max}^{\text{dn}} \cdot \beta_t, & \text{for } \beta_t < 0 \qquad (2.23) \end{cases}$$

After EA, we update the available power capacity:

$$P_{\max}^{\text{up}} = P_{\max}^{\text{up}} - P_t^{\text{EA}}, \text{for} \beta_t \geq 0 \tag{2.24}$$

$$P_{\max}^{\text{dn}} = P_{\max}^{\text{dn}} - P_t^{\text{EA}}, \text{for} \beta_t < 0 \tag{2.25}$$

The frequent deployment of BSS will induce the degradation of the battery, which is considered as the cost during the operation. The cost model of degradation in [39] is used, which assigns a constant marginal cost for battery charging/discharging:

$$C_t = c \cdot [(P_{\max}^{\text{dn},0} - P_{\max}^{\text{dn}}) + (P_{\max}^{\text{up},0} - P_{\max}^{\text{up}})] \tag{2.26}$$

where c is the depreciation cost coefficient (\$/MW), which depends on the investment cost of the BSS; C_t is the degradation cost at time step t. The depreciation cost here is originated from cycle degradation, which is related with the battery operation regime. The introduction of the depreciation cost/operating cost prevents the batteries from excessive deployment, which is closer to the actual operating environment of the batteries.

The SOC of BSS is time-coupled. SOC_{t+1} is dependent on SOC_t and discharge/charge behavior at hour t:

$$SOC_{t+1} = SOC_t - \frac{P_t^{\mathrm{f}} q_t}{U} + \frac{P_t^{\mathrm{pv,e}} \Delta h \cdot \eta_{\mathrm{ch}}}{U} + \frac{[\mathrm{sgn}(\beta_t)]^+ P_t^{\mathrm{EA}} \cdot \Delta h \cdot \eta_{\mathrm{ch}}}{U} + \frac{[\mathrm{sgn}(-\beta_t)]^+ P_t^{\mathrm{EA}} \cdot \Delta h}{\eta_{\mathrm{dis}} U} \tag{2.27}$$

where sgn is the sign function and $[\quad]^+$ is the rectified linear unit (ReLU) function. The adoption of these two function serves as the logic expression: when $\beta_t > 0$, the fifth term of (2.27) becomes zero; when $\beta_t < 0$, the fourth term of (2.27) becomes zero.

After T time intervals, the cumulative revenue over the whole scheduling cycle is obtained by summing up the net profit of PV-BSS at each hour:

$$B = \sum_{t=0}^{T-1} B_t^{\mathrm{EA}} + B_t^{\mathrm{pv}} + B_t^{\mathrm{f}} - C_t \tag{2.28}$$

B is a random variable considering the time-varying and random features of the PV generation and market signals. Hence, the ultimate goal of the DRL agent is maximizing the expected value of B.

2.3 Deep Reinforcement Learning

This section mainly describes how to generate the control signal of EMU to guide the optimal and safe CS of the PV-BSS. In the following of this section, the MDP of CS for PV-BSS is formulated, followed by the derivation of the battery control signals generated by one of the cutting-edge DRL algorithms, the PPO [51].

2.3.1 Markov Decision Process

The CS problem of PV-BSS can be modelled as MDP, which can be solved by the DRL algorithm. A finite horizon discounted MDP is characterized by a tuple $(\mathbf{S}, \mathbf{a}, \mathbf{P}, r, \gamma)$, where $\mathbf{S}$ is the state vector, $\mathbf{a}$ is the action vector, $\mathbf{P}$ is the state transition function, r is the reward function, and γ is the discount factor. $\mathbf{P}$ is dependent on the environment and partially described in (2.27).

The essential elements in the finite horizon discounted MDP corresponding to the CS of PV-BSS are defined as follows:

Algorithm 1 Safety Control Algorithm for PV-BSS

```
Input:
    System parameter: $SOC_0$, $\overline{SOC}$, $\underline{SOC}$, $P^{\text{up},0}_{\max}$, $P^{\text{dn},0}_{\max}$,$P^{\text{f,min}}$,$\eta_{\text{dis}}$, $\eta_{\text{ch}}$, $U$; Predicted solar power: $\overline{\boldsymbol{P}}^{\textbf{pv}}$; Market signal: $\boldsymbol{F}^{\textbf{lmp}}$, $\boldsymbol{F}^{\textbf{CCP}}$, $\boldsymbol{F}^{\textbf{PCP}}$, $\boldsymbol{\lambda}$, $\boldsymbol{q}$; Control signal: $\boldsymbol{\alpha}$, $\boldsymbol{\beta}$, $\boldsymbol{\xi}$
for each $t \in [0, T-1]$ do
    Calculate $SOC_t^{\text{up}}$ and $SOC_t^{\text{dn}}$ using (2.5).
    Perform frequency regulation:
    Dispatch FR capacity via (2.6) or (2.7) depending on the sign of $q_t$, impose the clip function, obtain $P_t^{\text{f}}$ via (2.9) or (2.10) considering $P^{\text{f,min}}$.
    Update $SOC_t^{\text{up}}$ and $P^{\text{up}}_{\max}$ via (2.11) and (2.13), respectively; or update $SOC_t^{\text{dn}}$ and $P^{\text{dn}}_{\max}$ via (2.12) and (2.14).
    The revenue of FR is calculated by (2.2).
    Allocate PV power:
    if $\overline{P}_t^{\text{pv}} > 0$ then
        Dispatch the PV power via (2.15), (2.16), (2.17), (2.18). The revenue of selling PV power is calculated by (2.3).
    end if
    Perform energy arbitrage:
    if $\beta_t \geq 0$ then
        Conduct purchasing with (2.19), (2.21), (2.22), and (2.24).
    else
        Conduct selling with (2.20), (2.21), (2.23), and (2.25).
    end if
    The revenue of EA is calculated by (2.4), calculate the degradation cost with (2.26), and update SOC of BSS using (2.27).
end for
Calculate the cumulative net revenue using (2.28).
```

1. The state vector is represented as:

$$\mathbf{S} = [SOC_t^{\text{up}}, SOC_t^{\text{dn}}, F_t^{\text{lmp}}, q_t, \chi_t, \overline{P}_t^{\text{pv}}] \tag{2.29}$$

   where $\chi_t = \varphi_t \cdot \left(\lambda_t F_t^{\text{PCP}} + F_t^{\text{CCP}}\right)$. The agent can access the current state of the BSS, the market signals from the regulation and energy market, and the available PV generation. Statistical evaluation are applied to these data to ensure the represent the representativeness [57]. It is noted that only the information available at the current time step is included in the state vector. This is restricted by the Markov property of MDP, in which the conditional probability distribution of future states of the process depends only upon the present state, not on the sequence of events that preceded it.

2. The action vector is represented by $\mathbf{a} = [\alpha_t, \beta_t, \xi_t]$, which corresponds to the capacity dispatch decisions for the stacked services. The control problem in this work is characterized by the continuous action space, which is more appropriate for controlling the battery.

3. The reward function is defined as:

$$r_t = B_t^{\text{EA}} + B_t^{\text{pv}} + B_t^{\text{f}} - C_t \tag{2.30}$$

   where r_t also represents the net profit of the PV-BSS at hour t. Compared with the existing works which are dependent on designing a delicate reward function, the reward function used in this work is more straightforward and easier to implement.

   The operation of the battery must satisfy the safety constraints, including the maximum/minimum SOC and maximum power capacity. If we directly apply the DRL to control the operation of the battery, one indispensable step is integrating penalties on constraint violation into the objective function. Despite the prevalence of the penalty function method, it is notorious for lacking a systematic method to determine the proper penalty coefficients. The small penalty coefficients may cause the constraint violation, while the large penalty coefficients will introduce significant errors and lead to the deterioration of the performance of the agents. The choice of the penalty coefficients is so important, and it may dominate the performance of the optimality and the convergence of the algorithm. In contrast, the reward function defined above is quite straightforward and easy to implement. It avoids introducing the penalties for the violation of constraints in the reward function. This is attributed to the serial strategy of the safety control algorithm to some extent. One significant benefit brought by adopting such a reward function is the excellent convergence performance of the PPO agent, which is verified in the case studies section.

2.3.2 Proximal Policy Optimization

PPO is a cutting-edge DRL algorithm developed in [51]. PPO can guarantee the safe exploration of the agent and make the full use of the available samples simultaneously. Moreover, PPO can tackle the continuous and multi-dimensional action space readily. Hence, along with the safety control algorithm proposed in section 2.2.3, PPO is an appropriate solution to the CS problem of the PV-BSS.

2.3.2.1 Preliminaries and Notation

The advantage function, which measures how much an action is better than others on average, is defined as:

$$A^{\pi,\gamma}\left(s_t, a_t\right) = Q^{\pi,\gamma}\left(s_t, a_t\right) - V^{\pi,\gamma}\left(s_t\right) \tag{2.31}$$

$$V^{\pi,\gamma}\left(s_t\right) = \mathbb{E}_{s_{t+1:\infty}}\left[\sum_{l=0}^{\infty}\gamma^l r_{t+l}\right] \tag{2.32}$$

$$Q^{\pi,\gamma}\left(s_t, a_t\right) = \mathbb{E}_{\substack{s_{t+1:\infty}\\ a_{t:\infty}}}\left[\sum_{l=0}^{\infty}\gamma^l r_{t+l}\right] \tag{2.33}$$

where $V^{\pi,\gamma}$ and $Q^{\pi,\gamma}$ are the value function and action-value function, respectively; One of the widely used estimation approaches for $A^{\pi,\gamma}$ is the temporal difference generalized advantage estimation [53].

In the context of the actor-critic type DRL algorithm, π_θ represents the agent's policy on choosing the action and is parameterized by θ. In other words, π_θ is the actor network, and it maps the observation received by the agent to the action. V_ϕ represents the value function network parameterized by ϕ. V_ϕ is also denoted as the critic network. Both π_θ

and V_ϕ are represented as multi-layer perceptrons (MLP) because of their powerful function approximation capability.

2.3.2.2 PPO

Utilizing importance sampling techniques, PPO derives a novel policy gradient expression, which makes it possible to update the policy network multiple times after collecting the trajectory set. This strategy improves the sample efficiency and training stability of PPO.

As an appreciable distinction, PPO employs a clipped surrogate objective function [51]:

$$\hat{J}^{\mathrm{PPO}} = \max_{\theta} \mathop{\mathbb{E}}_{s,a\sim\pi_{\theta_{\mathrm{old}}}} \left[\mathbb{L}\left(s, a, \theta_{\mathrm{old}}, \theta\right)\right] \tag{2.34}$$

$$\mathbb{L}\left(s, a, \theta_{\mathrm{old}}, \theta\right) = \min[\rho_t A^{\pi_{\theta_{\mathrm{old}}}}(s_t, a_t), \mathrm{clip}\left(\rho_t, 1-\epsilon, 1+\epsilon\right) A^{\pi_{\theta_{\mathrm{old}}}}(s_t, a_t)] \tag{2.35}$$

where ϵ is the hyperparameter which controls the permissible policy deviation; ρ_t is a ratio coefficient between the updated policy and the old policy and $\rho_t = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\mathrm{old}}}(a_t|s_t)}$. Specifically, the further the value of ρ_t deviates from one, the farther the updated policy is from the original policy.

The motivation of the adoption of (2.34) is that it can deter the drastic change of the policy network, which may deteriorate the performance of the PPO agent. To be specific, a clip function can be interpreted as a regularizer for the policy network. For example, assume $A^{\pi_{\theta_{\mathrm{old}}}}(s_t, a_t) > 0$, (2.35) can be reduced to:

$$\mathbb{L}\left(s, a, \theta_k, \theta\right) = \min\left(\rho_t, 1+\epsilon\right) A^{\pi_{\theta_{\mathrm{old}}}}(s_t, a_t) \tag{2.36}$$

in which the value of $\pi_\theta(a_t|s_t)$ will be increased during the update process. However, if $\pi_\theta(a_t|s_t) > (1+\epsilon)\pi_{\theta_{\mathrm{old}}}(a_t|s_t)$, the min operator will be in effect and forces the ρ_t to stay at $1+\epsilon$. Similarly, the clip function will enforce the minimum of ρ_t to be $1-\epsilon$ if $A^{\pi_{\theta_{\mathrm{old}}}}(s_t, a_t) < 0$.

According to (2.34), the parameters of the policy network (actor) θ are updated as follows in the PPO:

$$\theta_{\mathrm{new}} = \arg\max_{\theta} \mathop{\mathrm{E}}_{s,a\sim\pi_{\theta_{\mathrm{old}}}} \left[\mathbb{L}\left(s, a, \theta_{\mathrm{old}}, \theta\right)\right] \tag{2.37}$$

In the implementation phase, there are three steps to update the parameters of the policy network (actor). Firstly, calculate $\underset{s,a\sim\pi_{\theta_{\text{old}}}}{\mathrm{E}}\left[\mathbb{L}\left(s,a,\theta_{\text{old}},\theta\right)\right]$ based on the experience data collected in the agent-environment interaction. Herein the expectation operator E is usually approximated by the mean operator in practice using the Monte Carlo approximation. Afterwards, because the optimizer in the deep learning libraries is designed to minimize the loss function, we can regard $\underset{s,a\sim\pi_{\theta_{\text{old}}}}{\mathrm{E}}\left[-\mathbb{L}\left(s,a,\theta_{\text{old}},\theta\right)\right]$ as the loss function. Lastly, the gradient update process can be conducted using the chain rule, followed by updating parameters as shown in (2.37).

Another vital network to be learned is the value function network, which is updated via the regression:

$$\phi = \arg\min_{\phi}\left[\left(V_\phi - \hat{R}_t\right)^2, \left(\operatorname{clip}\left(V_\phi, V_{\phi_{\text{old}}}-\varepsilon, V_{\phi_{\text{old}}}+\varepsilon\right)-\hat{R}_t\right)^2\right] \tag{2.38}$$

where $\hat{R}_t$ is the reward-to-go: $\hat{R}_t = \sum_{t'=t}^{T} r_t\left(s_{t'}, a_{t'}, s_{t'+1}\right)$.

2.3.2.3 Stochastic Diagonal Gaussian Policies

To tackle the continuous action spaces, this work employs the stochastic diagonal Gaussian policy. To be specific, the output of the actor network π_θ is assumed to be the mean vector of the actions, which follow a multivariate normal distribution with the diagonal covariance matrix. The diagonal elements of the covariance matrix are the variances of each action.

When the PPO agent attempts to determine the action based on the observation, it depends on the sampling of the actions from the multivariate normal distribution:

$$\mathbf{a} = \mu_\theta(s) + \sigma_\theta(s) \odot \mathbf{x} \tag{2.39}$$

where $\mathbf{x}$ is the sample vector of a standard multivariate normal distribution; μ and σ are the mean and standard deviation of the action vector; $\odot$ is the element-wise product. The exploration of the PPO agent is achieved by sampling the Gaussian distribution in (2.39).

Based on the stochastic diagonal Gaussian policies, $\pi_\theta(a_t|s_t)$ can be derived as:

$$\pi_\theta(a_t|s_t) = \frac{1}{(2\pi)^{k/2}\prod_i^k \sigma_i}\exp\!\left(-\sum_{i=1}^{k}\frac{(x-\mu_i)^2}{2\sigma_i^2}\right) \tag{2.40}$$

2.3.2.4 Early Stop Mechanism

Even with the clipped surrogate objective function described above, it is still possible that the updated policy gets too far away from the old policy during the update process. One practical technique to prevent this phenomenon is the early stop mechanism based on monitoring the Kullback-Leibler (KL) divergence of the policy.

KL divergence calculates a score that measures the divergence of one probability distribution from another. The KL divergence for distributions P and Q of a continuous random variable can be defined as:

$$D_{\mathrm{KL}}(P\|Q) = \int_{-\infty}^{\infty} p(x) \log\left(\frac{p(x)}{q(x)}\right) dx \tag{2.41}$$

where $p(x)$ and $q(x)$ are the probability density functions (pdf) of P and Q, respectively.

Since the outputs of the actor networks are the normal distributions, in our case, preventing the updated policy from getting too far away from the old policy is equivalent to preventing the approximate KL divergence of these normal distributions from violating the upper limit $D_{\mathrm{KL,max}}$. The approximate KL divergence of outputs at time step t is defined as:

$$D_{\mathrm{KL}}(\pi_{\theta_{\mathrm{old},t}}\|\pi_{\theta,t}) = \log(\frac{\pi_{\theta_{\mathrm{old}}}(a_t|s_t)}{\pi_\theta(a_t|s_t)}) \tag{2.42}$$

where $\pi_\theta(a_t|s_t)$ here can be interpreted as the value of the PDF of action a_t under the state s_t and current policy π_θ.

During the process of updating the actor network within one epoch, once the average $D_{\mathrm{KL}}(\pi_{\theta_{\mathrm{old},t}}\|\pi_{\theta,t})$ over all time steps and all episodes exceeds the threshold $D_{\mathrm{KL,max}}$, the early stop mechanism will be activated to stop the further gradient updates.

2.3.2.5 Training Scheme

The goal of the PPO agent is maximizing the expected cumulative reward along the trajectory. The training scheme is summarized in Algorithm 2 [51].

2.4 Numerical Result

To demonstrate the effectiveness of the proposed method, case studies are conducted based on the real-world data from PJM [58]. The solar power data is obtained from the

Algorithm 2 PPO Agent Training Scheme

```
Initialize policy network $\pi_\theta$ and value function network $V_\phi$.
for $i = 0; i < N; i++$ do
    Policy agent $\pi_{\theta_{\text{old}}}$ interacts with the environment using (2.39) and records the trajectories samples $\{\tau_i\}$. Calculate the reward-to-go $\hat{R}_t$.
    Based on the value function $V_{\phi_{\text{old}}}$, perform the advantage estimation and obtain $A^{\pi_{\theta_{\text{old}}}}$.
    Update $\pi_\theta$ $N_\pi$ times with early stop mechanism using (2.34).
    Update $V_\phi$ $N_V$ times using (2.38).
    $V_{\phi_{\text{old}}} \leftarrow V_\phi$; $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$.
end for
```

National Renewable Energy Laboratory [59]. The parameters of PV-BSS to be used in the case study are summarized in Table 2.1. Table 2.1 also shows the hyperparameters of the PPO. N_π and N_V are the numbers of iterations of the actor network and critic network, respectively. lr_π and lr_V are the learning rates for the Adam optimizer of the policy network and the value function network, respectively. As the benchmark, A2C shares the same hyperparameters except for $KL^{\max}$, ϵ and N_π. Both PPO and A2C use the MLPs with two hidden layers as the policy network and the value function network, respectively. The number of neurons in each hidden layer is 64. The activation function of the hidden layer is a hyperbolic tangent function. The output activation function of the policy network is a hyperbolic tangent function as well. After obtaining the action vector $\mathbf{a} = [\alpha_t, \beta_t, \xi_t]$ from the policy network, α_t and ξ_t are mapped to a [0,1] space to produce the expected control signal.

In addition to the A2C agent, the DDQN agent adopted in [41] with the discrete action space is applied to the CS task. The action spaces of the DDQN are designed as suggested in [41]: $\alpha_t \in [0, 1/2, 1]$, $\xi_t \in [0, 1/2, 1]$, and $\beta_t \in [-1, -1/2, 0, 1/2, 1]$. Hence, the action dimension of the neural network, i.e., the size of the output layer, is set to be 3*5*3=45. Apart from PPO and A2C, another well-known DRL agent in the continuous action space is DDPG, which is characterized by learning the Q-function and policy simultaneously and its deterministic policy. One of the essential ideas in DDPG is that it approximates the

Table 2.1: Parameters of the PV-BSS and hyperparameters of the PPO

Parameters	Value	Parameters	Value
S_0	0.5	ϵ	0.2
ϕ	0.95	γ	0.91
$\eta_{\text{dis}}/\ \eta_{\text{ch}}$	0.9/0.9	λ	0.97
$\underline{S}\ /\ \overline{S}$	0.1/0.9	$lr_\pi\ /\ lr_V$	5.7e-4/1.2e-7
U	30 MWh	$N_\pi\ /\ N_V$	80/80
c	0.5$/MW	$KL^{\max}$	0.015
$P^{\text{dn}}_{\max}/\ P^{\text{up}}_{\max}$	-10MW/10MW		
$\Delta h\ /\ \Delta t$	1h / 4s	$\log \sigma_\theta$	-0.6

calculation of action which maximizes the Q function using the output of the policy network, which eliminates the need for solving a highly non-trivial optimization problem. As an opponent, the DDPG [60] agent is also implemented to conduct the CS task under the same environment as PPO. Consistent with the DDQN, the DDPG also uses the replay buffers and the target networks strategy to stabilize the training. The hyperparameters of the DDPG agent and the DDQN agent are well-tuned by using hyperparameters tuning technique to achieve the best performance [41].

The scheduling cycle is one week (168 h) in the case study. Thus, the predefined trajectory length is 168. The market data in 2018 are split into training and testing sets: the first nine months are for training, and the rest three months are the testing set. For each epoch, 12 trajectories are collected to update the PPO and A2C agents.

All tests are performed on a computer with Core i7 processor running at 3.2 GHz and 16 GB of RAM. The DRL code is implemented on the platform of Pytorch, and the hyperparameters tuning is performed using Optuna package on Python.

2.4.1 Performance of PPO

The average weekly revenue is regarded as an index to evaluate the learning performance of the DRL agents. Figure 2.4 shows the average weekly revenue evolution curves of the PPO, A2C, DDQN and DDPG agent during the training process. The PPO agent converges after epoch 75, reaching around $47,700. To evaluate the training time of the PPO, five different random seeds are used independently. it took a total of 300.75 +/- 2.85 s for the

PPO to reach convergence, while the entire training process (200 epochs) consumes 795.30 +/- 3.55 s in total.

The initial points of the curves represent the performance of the random policy, which is around \$30,168. Compared with the random policy agent, the PPO agent improves the net profit by about 58.1%. The random agents defined in our work are equivalent to the untrained agents. A PPO agent is made up of the policy and value networks, which are represented by two different MLPs. In our implementation, as suggested in [61], the orthogonal initialization and layer scaling techniques are applied to give the PPO agent a better initial policy and value network. One of the motivations for using such initialization is to speed up the convergence of the agent.

The A2C agent converges to \$36,355 after epoch 150. the PPO agent outperforms the A2C agent by about 31.2% in terms of revenue. The reason is that PPO employs a clipped surrogate objective function, which allows the approximately biggest possible improvement on the policy network every iteration, thereby avoiding the aggressive update and the performance collapse.

The DDQN agent converges to \$34,700 after epoch 160. The PPO agent outperforms the DDQN agent by about 37.5%, which justifies the necessity of the continuous action space for the CS of battery. The DDPG agent converges to \$39,880 after epoch 140. The plateaus shown in the DDQN and DDPG agent are caused by the mechanism that agents will not be trained until filling up the replay buffers. Even with the advantage of handling continuous action spaces and off-policy design, DDPG is still inferior to PPO agents for the CS problem in this work. The PPO agent outperforms the DDPG agent by about 19.6%. In addition, the training stability of PPO is significantly better than that of DDPG. The training curve of PPO is an approximately monotonically increasing curve; however, the curve of DDPG undergoes a deterioration of performance during training. After the training is completed, the performance of PPO is relatively stable in the interval of epoch [75,200] with a variance of 256.59. In contrast, the variance of the performance of DDPG in the interval of epoch [140,200] is 315.07, which is greater than that of PPO by 22.79%. This result can justify the superiority of the PPO agent over the DDPG agent in the training stability.

It can be observed that PPO provides a better convergence and performance rate than

DDPG. This is because DDPG is limited by: 1. Since Q values are very noisy, Q function network tends to overestimate the action values, causing the algorithm to converge to a poor solution; 2. there are four networks in the DDPG agent, namely, policy network, Q function network, target policy network, and target Q function network. The interaction between the current network and the target network makes convergence more difficult, compared to the PPO agent with only two networks; 3. To make DDPG policies explore better, noise upon the actions at the training phase is introduced. The convergence of the DDPG agent relies highly on the noise setting, whereas there is not a systematic way to determine the scale of the noise. What is more, the PPO agent directly optimizes for the agent performance, as opposed to the DDPG agent that trains the Q function to satisfy the Bellman equation. This feature also makes the PPO agent more stable and reliable [62, 63].

Another perspective for evaluating the efficiency of the DRL algorithms is sample efficiency. DRL is a class of trial and error learning methods. In the computation of the DRL algorithm, in addition to training the neural network, a lot of time is spent on the interaction between the agent and the environment, that is, on collecting experience. Therefore, the best performing agent with the smallest number of samples is preferable. PPO uses only about 36*12*168 = 72,576 samples to achieve the ultimate performance of A2C, where A2C needs 150*12*168 = 302,400 samples. Since DDPG and DQN are off-policy DRL algorithms, they use random sampling from inside the replay buffer for training, a mechanism that is significantly different from on-policy PPO. Therefore, there is no direct comparison between the sample efficiency of PPO and off-policy type algorithms. However, we can see from Fig. 2.4 that PPO still uses fewer epochs to achieve better performance.

In terms of testing data, Figure 2.5 presents the revenue of each week earned by the DRL agents from October to December in 2018. The performance upon the testing set can demonstrate the generalization of the DRL agent because the agents have never been exposed to these data before. It can be seen from Figure 2.5 that the PPO agent is dominant over the A2C agent, the random agent, the DDQN agent, and the DDPG agent all the time. To be specific, depending on the PPO agent, the PV-BSS can obtain $511,703 net profit in total from October to December, which is $20,606 more than the DDPG agent, $26,925 more than the A2C agent, $36,400 more than the DDQN agent and $212,128 more than the random agent. Hence, the PPO agent can adapt to the uncertain environment. The results above can demonstrate the superiority of the PPO agent over other DRL agents in addressing the CS problem of PV-BSS.

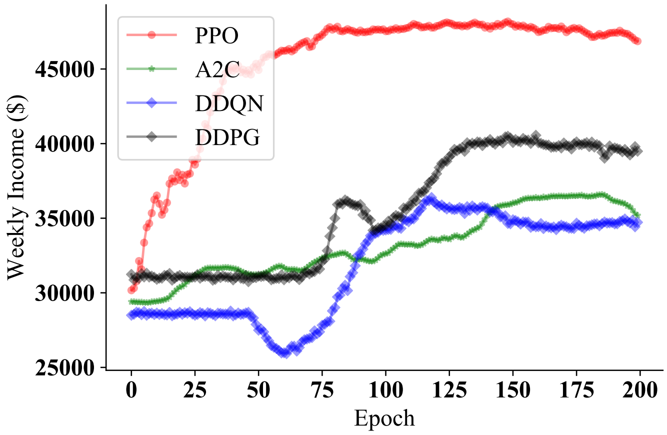


Figure 2.4: Training process of PPO and A2C.

2.4.2 Discussion of the Sequence of the Services

TABLE 2.2 summarizes the number of epochs needed for different service sequences to converge to the optimal solution (around $47,700). The numbers 1, 2, and 3 in the table represent the FR, PV charging, and EA, respectively. The data presented in the table is obtained from the different PPO agents with the well-tuned hyperparameters to report each agent's best performance and conduct a fair comparison.

The sequence of services does not impact the optimality of the DRL agent, which demonstrates the robustness of the DRL agent to seek long-term cumulative gains in a changing environment. However, the sequence of services will affect how quickly the algorithm converges. The sequence S123 is the most efficient among all the sequences, which takes only 75 epochs to converge. The rationale for this result is given as follows.

Assume that $q_t < 0$ and $P_t^{\text{EA}} > 0$ at hour t, the services which lead to the rising of the SOC include FR, PV charging, and EA. Through performing the FR, the PV-BSS can raise

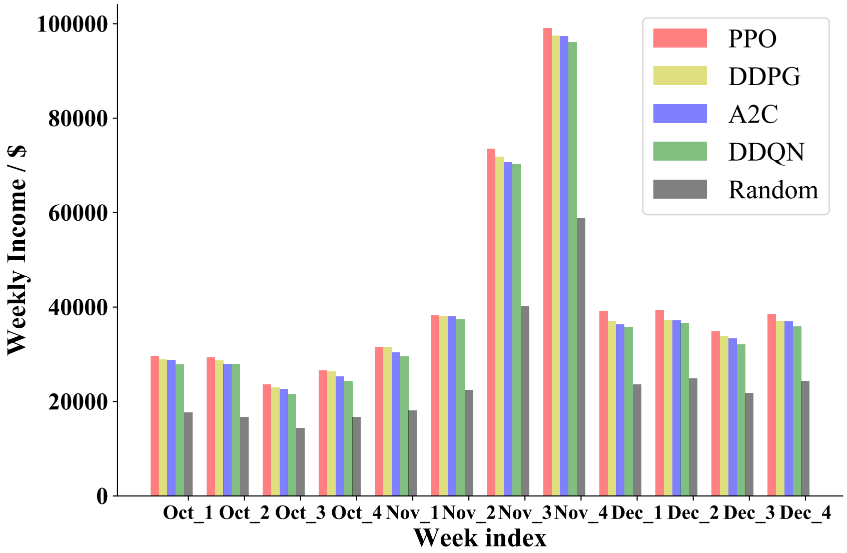


Figure 2.5: Revenue on the testing data.

Table 2.2: Number of epochs needed for different service sequences to converge to the optimal solution

Sequences	S123	S132	S213	S231	S312	S321
Epochs	75	149	294	512	330	273

the SOC and get paid as well. In contrast, to raise the SOC, performing EA requires the PV-BSS to purchase electricity from the energy market. As an intermediate, performing PV charging can raise the SOC at no expense. Hence, the DRL agent should decide P_t^{f} first. Afterward, based on the updated upward space of power capacity, the DRL agent settles $P_t^{\mathrm{pv,e}}$ and $P_t^{\mathrm{pv,s}}$, followed by P_t^{EA}.

Assume that $q_t > 0$ and $P_t^{\mathrm{EA}} < 0$ at hour t, which means performing the FR and EA will decrease the SOC of BSS. The energy-neutral characteristic of the RegD signal enable the PV-BSS to obtain the remuneration without losing much battery energy [58]. Besides, it is reported in [40, 55] that FR is the principal revenue source for BSS. Thus, herein the DRL agent should allocate P_t^{f} first, followed by P_t^{EA}.

2.4.3 Capacity Scheduling Scheme

The CS scheme for the first week of January is presented in Figure 2.6. The SOC curve of the battery is within the safety range of [$\overline{SOC}$, $\underline{SOC}$] over the whole scheduling cycle. Furthermore, the power capacity deployed to the stacked service is also within the safety constraint. This result verifies the effectiveness of the proposed safety control algorithm of PV-BSS to perform stacked services. The power capacity of BSS deployed for performing FR is dominant over all other services most of the time. This is because of the pay-for-performance mechanism of the PJM regulation market, which provides a significant economic incentive for the BSS with the fast-response feature. The PPO agent prefers to selling the PV power generated at the current hour instead of storing it because using the limited power capacity for FR and EA is more profitable in our case.

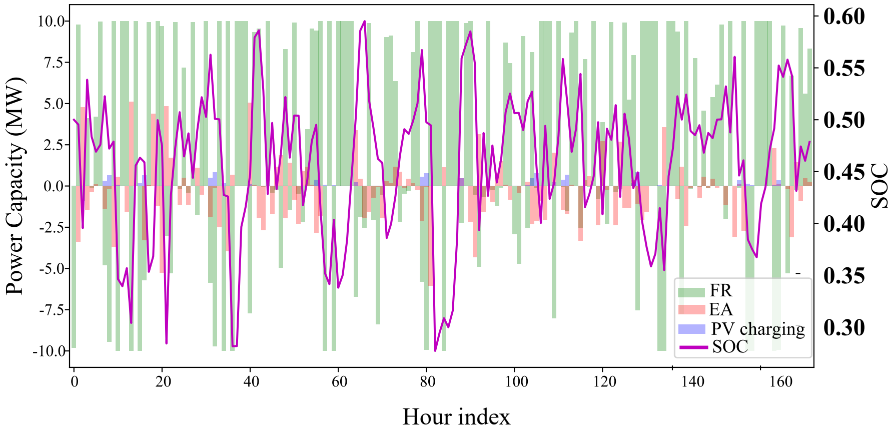


Figure 2.6: Capacity scheduling scheme for the first week of January.

2.4.4 Analysis of Revenue

The detailed revenue of the stacked services is shown in Figure 2.7. According to the statistics of the first week in January, the net profit is \$13195, which is made up of selling PV power(\$9,986), EA(\$1023), FR(\$2852), and depreciation cost(-\$666). It is noted that the calculation of the revenue and depreciation cost are based on the model presented in Section 2.2.2 and Section 2.2.3, respectively. It is easy to see that the revenue from the daytime is much more than that from the nighttime because of the availability of solar power. Besides, FR provides as much as around three times the profits of EA, which is consistent with their scheduled capacity.

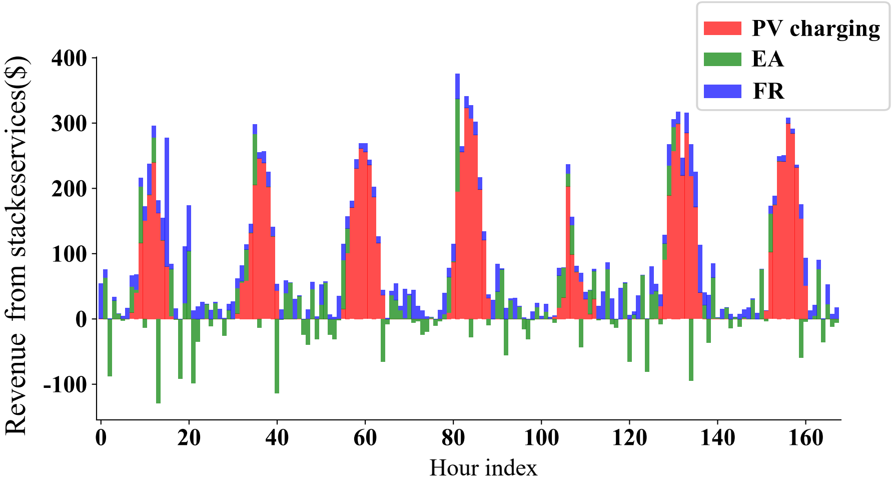


Figure 2.7: Detailed revenue and expenditure of the stacked services.

Figure 2.8 shows the power capacity deployed for EA and the LMPs in the first week in January. The result corroborates the capability of the PPO agent to make profits with EA. It can be observed that the PPO agent can capture the price trend of the energy markets and make a judicious decision. Most of the time, the PPO agent purchases at a relatively low price and sells at a relatively high price. It is noted that the price trend is not consistent with the EA power perfectly because of the existence of other services.

It is worth mentioning that the trained PPO agent consumes only 78 ms to make the scheduling decision, resulting from the computationally efficient feed-forward matrix computation of the neural network. The unrivalled execution speed reveals the potential of PPO in the real-time market environment.

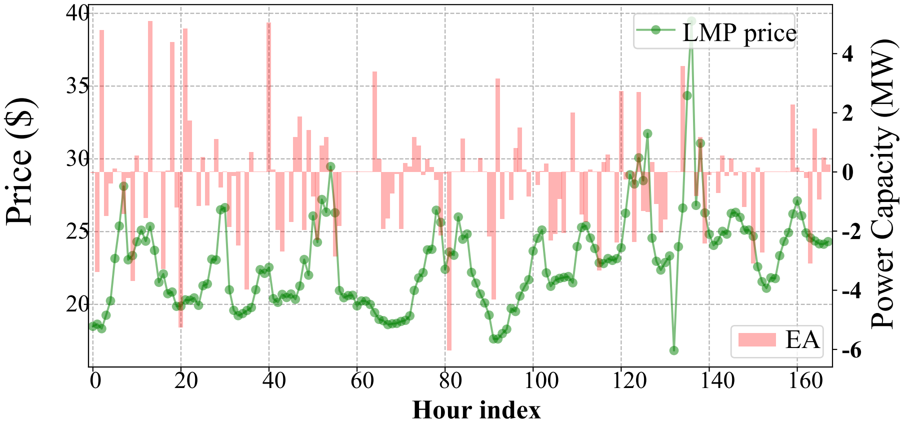


Figure 2.8: Power capacity deployed for EA and LMP price in the first week of January.

2.4.5 Analysis of Hyperparameters

Two popular hyperparameter optimization frameworks used in machine learning are Exhaustive Grid Search (EGS) and Randomized Parameter Optimization (RPO), respectively. The EGS exhaustively generates candidates from a grid of hyperparameters specified by the users. Afterwards, independent experiments are run on these candidates exhaustively to find the best hyperparameters. In contrast, RPO features searching randomly over hyperparameters, where each setting is sampled from a distribution over possible hyperparameter values. RPO is utilized here because of its flexibility which enables the search over a large range of hyperparameters without loss of efficiency. Besides, a budget can be chosen independent of the number of hyperparameters and possible values.

After numerous simulations, we find that three hyperparameters dominate the performance of the PPO agent in the context of the CS of battery, namely lr_π, lr_V, and γ. In addition, all other hyper-parameters are listed in TABLE 2.1.

lr_π and lr_V are the learning rates for the Adam optimizer of the policy network and the value function network, respectively. γ is the discount factor for evaluating the value function. Assume the lr_π and the lr_V follow the log-uniform distribution over (1e-7, 1e-4) and (1e-7, 1e-5), respectively. Assume the γ follows the uniform distribution over (0.9, 0.99). Considering the limited computational resources, 30 independent optimization calculations

Table 2.3: Hyperparameter tuning result

index	Weekly Net Revenue / $	γ	lr_π	lr_V
#4	30532	0.920	1.33E-07	1.99E-06
#5	31121	0.923	5.39E-07	8.32E-07
#8	31759	0.928	6.73E-07	4.21E-07
#2	34730	0.939	1.10E-06	2.69E-06
#6	34992	0.938	1.16E-06	5.17E-06
#9	36093	0.942	1.70E-06	1.40E-06
#15	39764	0.905	4.23E-06	3.69E-07
#17	39770	0.909	4.29E-06	1.26E-07
#19	41871	0.901	6.50E-06	6.20E-07
#26	45215	0.901	1.71E-05	1.36E-07
#20	45378	0.918	2.98E-05	2.27E-07
#23	45379	0.901	4.91E-05	1.07E-07
#16	45435	0.911	2.15E-05	1.91E-07
#18	45484	0.916	2.71E-05	1.86E-07
#14	45514	0.908	3.00E-05	3.93E-07
#22	46541	0.900	3.35E-05	1.44E-07
#21	47044	0.912	5.62E-04	1.21E-07

based on 30 different hyper-parameter setting samples are conducted.

TABLE 2.3 shows the performance of the best eight as well as the worst eight groups and their corresponding hyperparameter settings. We can summarize the following general pattern from the table: relatively small γ and lr_V help to improve the performance of the PPO agents in the given range above. Conversely, agents with relatively large lr_π perform better.

2.5 Summary

This work proposes a pragmatic solution to the capacity scheduling of PV-BSS, which performs the stacked services. A safety control algorithm of PV-BSS is proposed to ensure the safe operation of the PV-BSS. A PPO-based DRL agent is developed to cooperate with the control algorithm to improve the profitability of PV-BSS. Case studies based on the real data of the PJM energy and regulation markets are conducted. In the training phase, the PPO agent outperforms the DDPG agent, the A2C agent, the DDQN agent, and the random policy agent by 19.6%, 31.2%, 37.5%, and 58.1% in terms of the weekly net profit, respectively. Moreover, the PPO agent is significantly more sample-efficient than the A2C agent. The PPO agent also shows better adaptivity than other DRL agents throughout the test set. The results on the testing data verify the PPO agent can adapt to volatile market signals and PV generation scenarios. Case studies on the real-world data demonstrate that the PPO agent is capable of generating safe scheduling schemes while maximizing the net profit of PV-BSS.

Chapter 3

Adaptive Static Equivalences for Active Distribution Networks with Massive Low-Carbon Energy Integration: A Distributed Deep Reinforcement Learning Approach

This section delves into the challenges of maintaining data fidelity in equivalent models for active distribution networks with 100% renewable energy sources penetration. By introducing an adaptive equivalent model for active distribution networks with tunable scales and parameters, and employing a leaves-trimming network topological reduction alongside a distributed Proximal Policy Optimization-based deep reinforcement learning agent, the work addresses the volatility of renewable energy sources and their management techniques. The approach demonstrates enhanced accuracy and efficiency in scenarios with substantial photovoltaic and wind generation, particularly when renewable energy production surpasses load demand.

3.1 Introduction

A decarbonized power sector is an indispensable path to tackle the net-zero emissions undertaking by 2050. The renewable energy transition, such as the massive integration of wind and solar, can significantly mitigate greenhouse-gas emissions, thereby contributing to electricity decarbonization [64]. This study attempts to develop an artificial intelligence-empowered modeling approach for an emerging form of low-carbon distribution networks, i.e., active distribution networks (ADNs) [65].

ADNs are characterized by large-scale renewable energy sources (RES) integration and active management of distribution resources [66]. The increasing number of heterogeneous components and the advanced control strategies further convolute the operating status of ADNs. Generally speaking, the distribution networks connect to the transmission network through the distribution substation. The proliferation of the RES integration will result in reverse power flow, which will raise the concern of the transmission system operator (TSO) and drive the TSO to appraise the impact of ADN on reliability and security. Some ADNs, organized in the form of microgrids, can be networked to share redundant reserve resources, thereby improving the reliability of power supply and guaranteeing the power quality [67]. In these two scenarios, the ADN and its interconnected objects are operated by different entities. Due to the concern of preserving sensitive information and saving computation resources, it is unlikely and unnecessary for the interconnected objects of ADN to grasp the detailed model of ADN. In some distribution networks with poor observability, even the distribution system operators (DSOs) can not maintain an accurate model of the network in real-time. Equivalent modeling techniques are practical solutions to the issues above. Moreover, some computationally intensive tasks (e.g., quasi-steady-state time simulations) on ADNs can be significantly accelerated if a simpler but accurate surrogate model is available. All the above factors jointly motivate this paper to develop a novel equivalent model of ADNs (EMADNs), which can take the characteristics of ADNs into account.

Revolving around the distinctions of distribution networks, some specific equivalencing models for distribution networks (DEMTs) have been surfacing [68–72] in recent years. [68] has examined the DEMT based on holomorphic embedding, thereby maintaining the accuracy of equivalent models when scaling the loads and the active power generations. However, the implicit expression of the equivalent models and the dependency upon the specific load flow algorithm restricts the scope of application. [73] builds a reduced network synthesis for ADNs by developing an improved Kalman filter to process the time series from phasor measurement units (PMUs). A two-tier DEMT that establishes the equivalent model based on the nodal voltage equations is established in [69]. The parameters of the equivalent generator and equivalent lines are calculated in the first stage, and the detailed parameters of equivalent load are obtained by least square estimation in the second stage. [70] develops the equivalent circuits through the inversion of the nodal admittance matrix of the load flow equation. [71] extends [70] to consider the aggregation of voltage-controlled devices, which improves the accuracy of EMADNs in terms of reactive power output. [72] uses grey box theory to construct the EMADNs with large-scale rooftop photovoltaic (PV) systems embedded, in which the aggregation of PV is treated as an independent entity rather than a negative load.

Albeit the DEMTs introduced in references [68–77] have been making significant advances, there are still some certain knowledge gaps to be filled. First and foremost, the large-scale integration of RESs brings significant uncertainty to ADNs, spawning the volatile operating points of the systems. Prior works on EMADNs [69, 72, 78] attempt to develop equivalent models by formulating a constrained quadratic optimization problem based on the historical measurements. This solution paradigm expects the derived parameters to maintain effectiveness under various conditions. Such assumptions are prone to inaccuracy considering the large-scale penetration of RESs. In contrast, using the available measurement/forecast data to adjust the established EMADN paves a path to address this issue. In addition, [66,69,72,78,79] neglect the topology of ADNs, which fails to represent the interior patterns of ADNs, including nodal voltages and line flows. Another hurdle is the availability of sufficient data to establish the EMADN. For instance, [70] and [71] are dependent on the exhaustive ADN model, e.g., every RES generation/load profile, voltage measurements, and up-to-date line parameters. However, this assumption does not hold in some distribution networks due to the prohibitive cost of the large-scale deployment of advanced measurement units. Lastly, the increased complexity of ADNs induced by the common voltage regulation schemes (VRS) and inverse power flows is not fully addressed.

This study casts the construction of EMADNs to a Markov Decision Process (MDP) and proposes a topological reduction algorithm. The motivation behind the MDP formulation can be summarized as 1. this study attempts to develop an accurate EMADN over a period of time (e.g., one day), during which the outputs of RES and load may fluctuate dramatically. As a discrete-time stochastic control process, MDP can incorporate this type of randomness readily; 2. Formulating the equivalent parameter identification as a sequential decision-making problem is conducive to making the optimal long-term decisions by considering the temporal correlation of system states. As an organic synthesis of deep learning (DL) and reinforcement learning (RL), deep reinforcement learning (DRL) is good at tackling sequential decision-making problems under uncertainty [80, 81]. [82] reviews the advancements of DRL on power system parameter identification, where DRL has already served as a state-of-art tool. [83] utilizes the double deep Q-learning for calibrating the load model parameters. The generator parameters are maintained automatically using Soft Actor-Critic and PMU measurements in [84]. A damping controller for low-frequency oscillation is tuned by the DRL agent in [85]. [86] and [66] attempt to construct the equivalent models for regional grids by using conventional RL as a heuristic function optimizer, which is prone to the curse of dimensionality of states and actions. More importantly, the adopted constrained quadratic optimization paradigm suffers from inadaptability. In contrast, with moderate requirements for measurements/forecasts, this study attempts to solve the intractable EMADN problem by leveraging the function approximation of the neural network and the model-free RL algorithm jointly, i.e., DRL, for the first time. Distributed learning scheme [87] is devised and adapted to accelerate the learning and diversify the training samples.

The DSO can conduct the training of the EMADNs. Once trained, DSO can use the model alone or share the trained models with the interconnected interested parties (TSOs or other DSOs). Users of the proposed EMADNs can update model parameters online with the measurements/forecasts of the net amount of aggregated RESs and loads. The requirement for individual RES and load data is waived. Notably, it is demonstrated in the subsequent case studies that the excellent fidelity of the proposed intelligent EMADNs can extend to scenarios in which there is a surplus production of low-carbon energy. This feature makes the proposed method a promising tool for appraising the feasibility of the massive low-carbon energy installation.

Our proposed method offers the following unique features and contributions:

1. Unlike prior work based on the fictitious topology and black box, a leaves-trimming network topological reduction algorithm is proposed in Section 3.3 to preserve the nodes of interest and radial topology of ADNs. The reduction method is achieved through the iterative eliminations of leaf nodes and can customize the reduction degree to meet the different requirements of model granularity.

2. To address the uncertainty caused by RESs and loads and leverage their temporal correlations, this study formulates the parameter identification of the EMADNs as an MDP in Section 3.4. The reward function of the MDP features the imitation capability of EMADNs on the boundary power exchanges to the external grids and on the nodal voltage magnitude.

3. A model-free DRL agent called Proximal Policy Optimization (PPO) is proposed in Section 3.5.1 to address the formulated MDP and build the EMADNs. The superiority of the DRL to the commonly used constrained quadratic optimization paradigm lies in the adaptivity of the equivalent models in a volatile environment. The proposed EMADNs feature the encapsulation and preservation of the radial typologies and the droop-based VRSs, which significantly impact the accuracy of voltage and reactive power.

4. A distributed training framework for the PPO agent (DPPO) is developed in Section 3.5.2 to alleviate the bottleneck of training efficiency of the DRL agent by performing the agent-environment interactions and gradient computation in a parallel manner. The data parallelism and the average gradient technique are integrated into the DPPO seamlessly.

The rest of the study is organized as follows. Section 3.2 describes the essential components and techniques of the ADNs. Section 3.3 proposes a leaves-trimming network topological reduction algorithm to provide a backbone topology for the EMADNs. The MDP for the parameter identification of EMADNs is formulated in Section 3.4, followed by the distributed training scheme of the DPPO algorithm in Section 3.5.1. Numerical studies are conducted in Section 3.6 to demonstrate the practicality and accuracy of the proposed agent-based EMADNs. Finally, concluding remarks are given in Section 3.7.

3.2 Models of ADNs

This section introduces an overview of the ADN and the proposed EMADN, followed by the models of voltage-dependent loads and low-carbon energy production equipped with the droop-based VRSs in ADNs.

3.2.1 Overview

The IEEE 33-bus ADN is used for the convenience of illustration in this section, which is shown in Fig.3.1. Among all 33 nodes, nine nodes are equipped with RESs, including eight PV generators and one wind turbine. All nodes are connected to loads except the root node 1, which connects to the external grid via the distribution substation.

Fig. 3.1 shows an illustrative example of the proposed equivalent model for the 33-bus system. The derivation of the proposed equivalent model will be elaborated in this section and Section 3.3. Compared to the previous network, the proposed EMADN features fewer branches, fewer nodes, simpler topology, and therefore fewer components. Nevertheless, the radial topology, specific nodes, and RESs with VRSs are well preserved and encapsulated. With the simplified structure and components, the proposed EMADN is an ideal tool for some computation-intensive applications. One example is evaluating the hosting capacity for the low-carbon energy installation in ADNs using scenario-based methods. The other is the security assessment of the transmission network considering the inverse power flow from ADNs using quasi-steady-state time simulations.

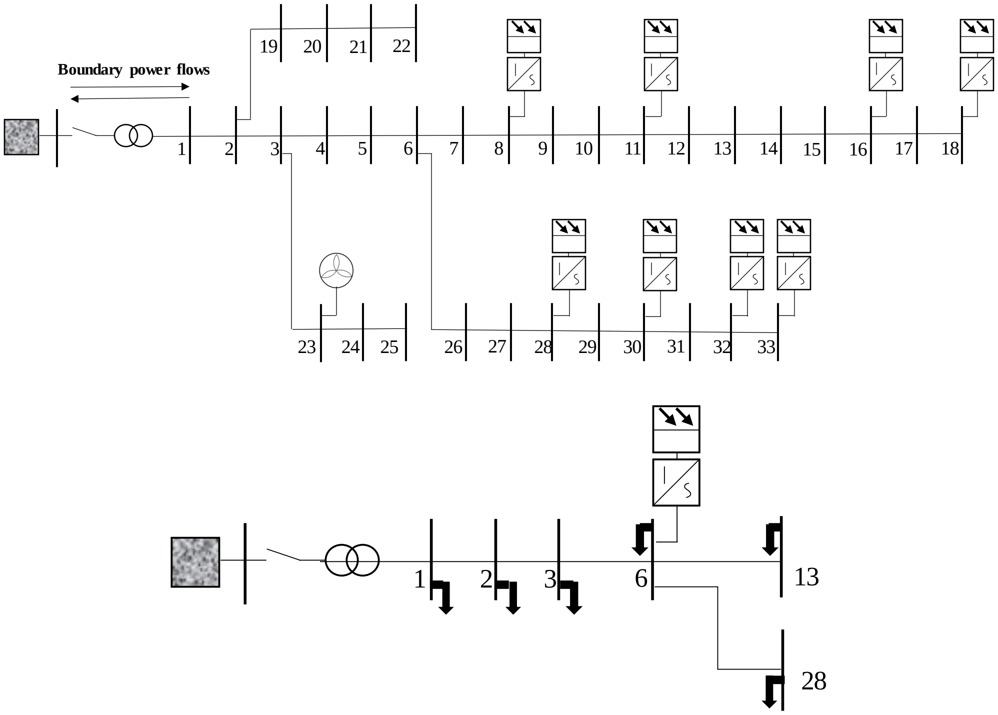


Figure 3.1: IEEE 33-bus active distribution network and an illustrative example of the proposed equivalent model.

3.2.2 Voltage-Dependent Loads

As a common model to characterize the voltage-dependent behavior of loads, ZIP loads are widely adopted. This study employs ZIP loads because of the inherent voltage characteristic of the composite load model for all classes of load types (industrial, commercial and residential). And this characteristic is more significant in the ADNs due to the volatile voltage operating points caused by RESs.

The ZIP load is represented as a nonlinear polynomial equation that is comprised of constant impedance (Z), constant current (I), and constant power (P) loads, respectively:

$$P_{\mathrm{load}} = P_{\mathrm{load,0}}(a_{\mathrm{Z}}^{\mathrm{p}} V_{\mathrm{m}}^2 + a_{\mathrm{I}}^{\mathrm{p}} V_{\mathrm{m}} + a_{\mathrm{P}}^{\mathrm{p}}) \tag{3.1}$$

$$a_{\mathrm{Z}}^{\mathrm{p}} + a_{\mathrm{I}}^{\mathrm{p}} + a_{\mathrm{P}}^{\mathrm{p}} = 1 \tag{3.2}$$

where P_{load} and $P_{\mathrm{load,0}}$ are the actual and rated active power demand of the load, respectively. a_{Z}, a_{I}, and a_{P} are the impedance, current, and power coefficients for the ZIP loads, respectively, which are constrained by (3.2). V_{m} is the nodal voltage magnitude. The model for reactive power loads ($Q_{\mathrm{load}}/Q_{\mathrm{load,0}}$) can be formulated in a similar manner.

3.2.3 RESs with Droop-based Voltage Regulation Schemes

According to IEEE standard 1547-2018 [88], voltage magnitude/reactive power (Volt-Var) droop control is a mandatory strategy for the interface of the inverter-based RESs to realize the distributed voltage regulation. Fig.3.2 shows the droop-based voltage regulation scheme (DVRS), which is represented as a piece-wise linear curve. The inverter will adjust the reactive power outputs of RESs according to the voltage at the feed-in point:

$$Q_{\mathrm{droop}} = \begin{cases} -\xi(V_{\mathrm{m,th,2}} - V_{\mathrm{m,th,1}}) & V_{\mathrm{m}} < V_{\mathrm{m,th,1}} \\ -\xi\left(V_{\mathrm{m,th,2}} - V_{\mathrm{m}}\right) & V_{\mathrm{m,th,1}} \leq V_{\mathrm{m}} \leq V_{\mathrm{m,th,2}} \\ 0 & V_{\mathrm{m,th,2}} < V_{\mathrm{m}} < V_{\mathrm{m,th,3}} \\ \xi\left(V_{\mathrm{m}} - V_{\mathrm{m,th,3}}\right) & V_{\mathrm{m,th,3}} \leq V_{\mathrm{m}} \leq V_{\mathrm{m,th,4}} \\ \xi(V_{\mathrm{m,th,4}} - V_{\mathrm{m,th,3}}) & V_{\mathrm{m}} > V_{\mathrm{m,th,4}} \end{cases} \tag{3.3}$$

where $V_{\mathrm{m,th}}$ are the voltage thresholds, ξ is the slope of the curves, and Q_{droop} is the feed-in reactive power. The proposed EMADNs incorporate this DVRS to capture the Volt-Var characteristic of the low-carbon energy. There are also other VRSs in the standard [88]. For example, in the active power-reactive power scheme, the reactive power outputs of the RESs are determined by their active power outputs. The DVRS employed in the proposed EMADNs can represent this scheme in an implicit way. This is because a direct consequence of the increase/decrease of active power injections is the increase/decrease of voltage magnitudes, followed by triggering the DVRS to control the reactive power outputs. As for the constant reactive power scheme and constant power factor scheme, the RESs operating under these schemes can be considered as conventional PQ loads in the proposed EMADNs. It is worth mentioning that the EMADNs proposed in this study can also be readily extended to other inverter control schemes depending on measurements.

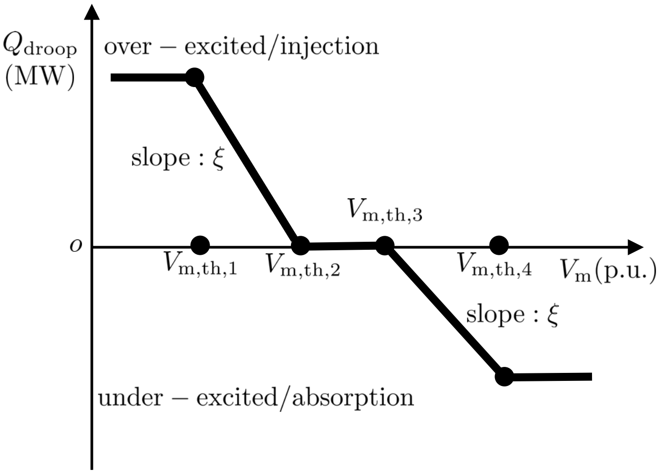


Figure 3.2: Response curve of the droop-based voltage regulation schemes.

3.3 Leaves-trimming Network Topological Reduction Method

There are three main challenges in the construction of the EMADNs: topology construction, determination of component locations, and parameter identification of components. A leaves-trimming network topological reduction method is proposed in **Algorithm 3** in this section to address the first two challenges. The last challenge is left to Section 3.4.

Algorithm 3 can 1). maintain the skeleton of the original system; 2). provide enough flexibility. Users can specify their interested nodes to retain and customize the degree of network reduction; 3). retain some real nodes in ADNs instead of introducing fictitious nodes. **Algorithm 3** assumes the radial topology and the locations of the RESs of the original system are available. Numerous studies have investigated the topology identification of ADNs, and a recent example can be found in [89].

The radial distribution network can be abstracted as a tree G in graph theory with a node set $\boldsymbol{N}$ and an edge set $\boldsymbol{E}$. In a nutshell, the core idea of the algorithm is to iteratively cut off the leaf nodes $\boldsymbol{N}_{\text{leaf}}$ in the tree, which is realized in Phase I. Then, in the second phase, all branches between the remaining leaf nodes and bifurcation nodes $\boldsymbol{N}_{\text{bif}}$ are aggregated into an equivalent branch to reduce the network scale further.

The presentation of **Algorithm 3** is self-explanatory. An example of the IEEE 33-bus system is adopted herein to gain a detailed understanding of it. In step 2, $\boldsymbol{N}_{\text{bif}} := \{2, 3, 6\}$, $\boldsymbol{N}_{\text{leaf}} := \{18, 22, 25, 33\}$, and $\boldsymbol{N}_{\text{int}}$ is assumed to be a null set. $\boldsymbol{N}_{\text{bif}}$ are essential nodes that reflect the radial topology of ADNs, and they will be retained during the whole reduction process. Moving on to step 3, when $i = 0$, the first reduction round is started. First, the RESs with the DVRS at nodes 18 and 33 are relocated to nodes 17 and 32 during steps 4 to 8, followed by removing all nodes in $\boldsymbol{N}_{\text{leaf}}$ and lines associated with them (17-18, 21-22, 24-25, and 32-33). Then, the leaf nodes and the remaining node set are updated accordingly in step 10. The rest of the loop w.r.t i can be completed similarly. The remaining nodes and edges after Phase I are shown in 3.3, in which different nodes are colored in different colors. During steps 12 to 14, for the leaf node 13, its nearest upstream bifurcation node is 6, so as the leaf node 28. Hence, two equivalent edges, i.e., 6-13 and 6-28, are created, along with aggregating the RESs with the DVRS in two paths (6 to 13 and 6 to 28) to node 6. During steps 15 to 21, for the bifurcation node 6, its upstream partner is node 3. Thus, an equivalent edge 3-6 is created. An equivalent edge 2-3 is generated under a similar process. On the other hand, since the bifurcation node 3 has no corresponding upstream bifurcation node, it is directly connected to the root node through the equivalent line 1-3. Fig. 3.1 shows an illustrative example of the proposed equivalent model after Phase II. Note that if more than one RES with the DVRS is placed in one node, they will be represented by an ensemble RES with the DVRS. The wind turbine in node 23 is aggregated as a load in node 3 because wind turbines usually run at fixed power factor mode, and they can be modeled as negative loads.

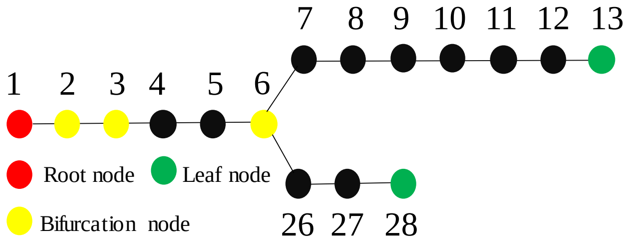


Figure 3.3: Remaining topology of IEEE-33 bus system after phase I of Algorithm 3 with $\beta = 5$.

3.4 Markov Decision Process for Parameter Identification

Once the topology and the locations of the components of EMADNs are determined, the next step is parameter identification [90]. This section casts the parameter identification of the EMADNs to a discrete stochastic control process, which can be modeled as an MDP. The crucial components of the MDP, i.e., state, action, and reward, are introduced successively.

3.4.1 State

A schematic diagram for the MDP and the DRL agent-environment interaction is shown in Fig.3.4. The environment is the world that the agent lives in and interacts with. Specifically, the environment is formulated as the MDP here. At every interaction, the agent observes the state of the world and then decides on an action to take. Not only does the environment evolve when the agent's actions take effect, but it may also evolve on its own. The agent receives the reward signal from the environment as feedback and leverages it to improve the decision-making capability.

The state is supposed to reflect the low-carbon energy adequacy of the ADNs while being subjected to the availability of measurement devices and forecast tools. Herein the total active/reactive power consumption of loads ($P_t^{\text{load}}/Q_t^{\text{load}}$) and the active power generation (P_t^{RES}) of total RESs are incorporated in the state vector. This assumption is moderate

Algorithm 3 Leaves-trimming Network Topological Reduction

```
 1: Input $G(\boldsymbol{N}, \boldsymbol{E})$, the root node $n_{\text{root}}$, the nodes of interest $\boldsymbol{N}_{\text{int}}$, the RES locations, and the customized
    reduction degree parameter $\beta$.
    **Phase I: backbone nodes identification**
 2: Identify the bifurcation nodes $\boldsymbol{N}_{\text{bif}} := \{n | n \in \boldsymbol{N}, \text{degree}(n) \geq 3\} \cup \boldsymbol{N}_{\text{int}}$; The leaf nodes $\boldsymbol{N}_{\text{leaf}} := \{n | n \in$
    $\boldsymbol{N}, \text{degree}(n) = 1, n \neq n_{\text{root}}, n \notin \boldsymbol{N}_{\text{int}}\}$; $G^* := G, \boldsymbol{N^*} := \boldsymbol{N}$
 3: for $i = 0; i < \beta; i++$ do
 4:    for every node $n$ in $\boldsymbol{N}_{\text{leaf}}$ do
 5:       if $n$ is connected to the RES with the DVRS then
 6:          Move the RES to the father node of $n$
 7:       end if
 8:    end for
 9:    Remove $\boldsymbol{N}_{\text{leaf}}$ and edges connecting these nodes from $G^*$.
10:    Update the leaf nodes as in Step 1. Update the remaining node set of $G^*$ to $\boldsymbol{N^*}$
11: end for
    **Phase II: creation of the reduced graph $G_{\text{eq}}$**
12: for every node $n$ in $\boldsymbol{N}_{\text{leaf}}$ do
13:    Create equivalent edge $l$ between $n$ and its nearest upstream bifurcation node $n_{\text{bif}}^{\{n\}}$ and add $l$ to $G_{\text{eq}}$;
       Move the RES generation between $n$ inclusive and $n_{\text{bif}}^{\{n\}}$ exclusive to $n_{\text{bif}}^{\{n\}}$.
14: end for
15: for every node $n$ in $\boldsymbol{N}_{\text{bif}}$ do
16:    if $n$ has a upstream bifurcation node then
17:       Create equivalent edge $l$ between $n$ and its nearest upstream bifurcation node $n_{\text{bif}}^{\{n\}}$ and add $l$ to
          $G_{\text{eq}}$; Move the RES generation between $n$ exclusive and $n_{\text{bif}}^{\{n\}}$ exclusive to $n_{\text{bif}}^{\{n\}}$.
18:    else
19:       Create equivalent edge $l$ between $n$ and $n_{\text{root}}$ and add $l$ to $G_{\text{eq}}$; Move the RES generation between
          $n$ exclusive and $n_{\text{root}}$ exclusive to $n$.
20:    end if
21: end for
```

compared to the assumption that requires the forecast data for each RES/load adopted in [70, 71].

In addition, to reveal the voltage and reactive power conditions of the ADNs, both the maximum and the minimum nodal voltage magnitudes are included in the state vector, which are denoted as $V_{\mathrm{m}}^{\max,t}$, $V_{\mathrm{m}}^{\min,t}$, respectively. These two variables are also accessible, which can either be acquired by the voltage measurement devices installed at the key nodes of ADNs, or by using the existing mature state forecasting methodology [91]. To summarize, the state vector of the MDP is defined as $\boldsymbol{s_t} := (P_{\mathrm{load}}^{t}, Q_{\mathrm{load}}^{t}, P_{\mathrm{RES}}^{t}, V_{\mathrm{m}}^{\max,t}, V_{\mathrm{m}}^{\min,t})$. Note that the state vector could be extended to include more informative states when more advanced measurement devices and forecast tools are available. However, this study attempts to use as little information as possible to evaluate and emphasize the decision-making capability of the agent.

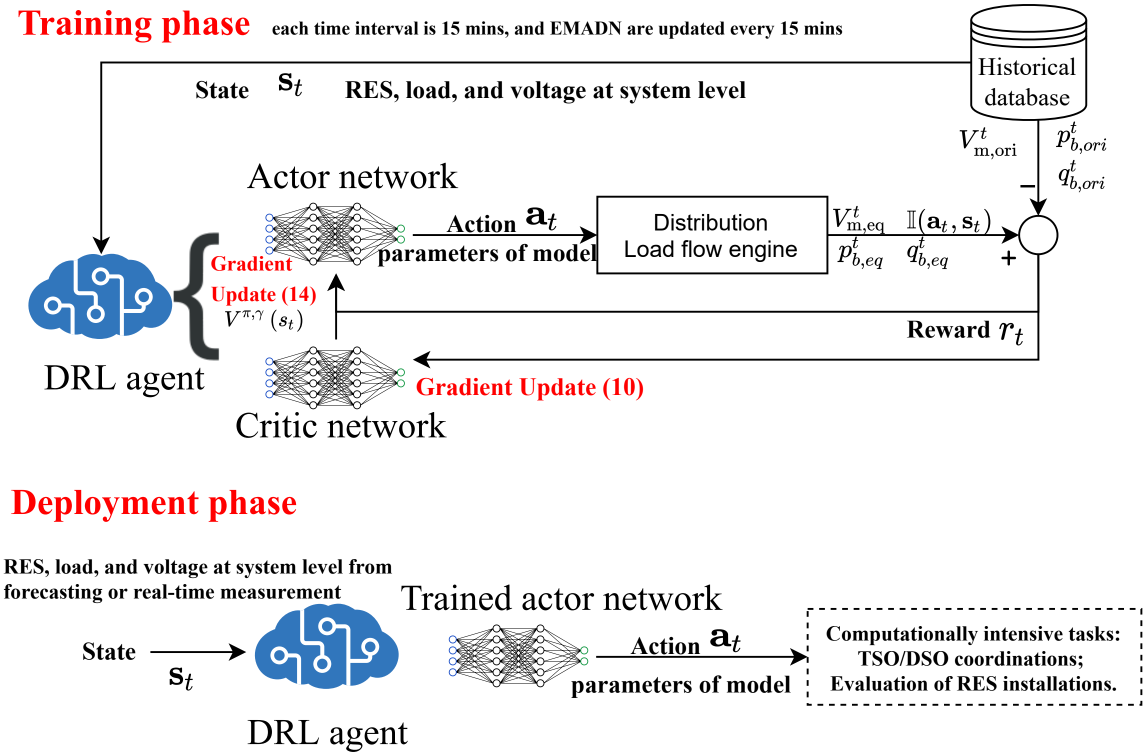

Training phase
each time interval is 15 mins, and EMADN are updated every 15 mins
Historical database
State $\mathbf{s}_t$ RES, load, and voltage at system level
$V_{\mathrm{m,ori}}^{t}$
$p_{b,ori}^{t}$
$q_{b,ori}^{t}$
Actor network
Action $\mathbf{a}_t$
parameters of model
Distribution Load flow engine
$V_{\mathrm{m,eq}}^{t}$ $\mathbb{I}(\mathbf{a}_t, \mathbf{s}_t)$
$p_{b,eq}^{t}$ $q_{b,eq}^{t}$
−
+
Gradient Update (14)
$V^{\pi,\gamma}(s_t)$
Reward r_t
DRL agent
Gradient Update (10)
Critic network
Deployment phase
RES, load, and voltage at system level from forecasting or real-time measurement
State $\mathbf{s}_t$
DRL agent
Trained actor network
Action $\mathbf{a}_t$
parameters of model
Computationally intensive tasks: TSO/DSO coordinations; Evaluation of RES installations.

Figure 3.4: Schematic diagram for the MDP and the DRL agent-environment interaction in training and deployment phase.

3.4.2 Action

The actions of the agent are defined as the parameters of the EMADN. Corresponding to Fig.3.1, the action vector is parameterized by $\boldsymbol{a_t} := (\ \boldsymbol{a}^t_{\mathrm{Pl}},\ \boldsymbol{a}^t_{\mathrm{Ql}},\ \boldsymbol{a}^t_{\mathrm{Z}},\ \boldsymbol{a}^t_{\mathrm{I}},\ \boldsymbol{a}^t_{\mathrm{R}},\ \boldsymbol{a}^t_{\mathrm{X}},\ \boldsymbol{a}^t_{\mathrm{Pg}},\ \boldsymbol{a}^t_{\xi},\ \boldsymbol{a}^t_{\mathrm{V_{th}}})$. $\boldsymbol{a}^t_{\mathrm{Pl}}$, $\boldsymbol{a}^t_{\mathrm{Ql}}$ are the equivalent active/reactive load baselines, respectively. $\boldsymbol{a}^t_{\mathrm{Z}}$ and $\boldsymbol{a}^t_{\mathrm{I}}$ are the ZIP load coefficients. $\boldsymbol{a}^t_{\mathrm{R}}$ and $\boldsymbol{a}^t_{\mathrm{X}}$ are the equivalent line resistance/reactance, respectively. $\boldsymbol{a}^t_{\mathrm{Pg}}$ is the active power of the RESs. $\boldsymbol{a}^t_{\xi}$ and $\boldsymbol{a}^t_{\mathrm{V_{th}}}$ are the DVRS parameters of the RESs.

3.4.3 Reward

The design of the reward function should reflect the user's expectations of the agent, which is equivalent to the fidelity of the EMADN in our case. The reward function is formulated as the summation of multiple penalty terms:

$$\begin{aligned} r_t(\boldsymbol{a}_t, \boldsymbol{s}_t) = -\mathrm{P}_1 * \mathbb{I}(\boldsymbol{a}_t, \boldsymbol{s}_t) - \sum_{i=1}^{N_{\mathrm{re}}} \mathrm{P}_2 |V^t_{\mathrm{m,ori},i} - V^t_{\mathrm{m,eq},i}| \\ - (p^t_{\mathrm{b,ori}} - p^t_{\mathrm{b,eq}}(\boldsymbol{a}_t))^2 - (q^t_{\mathrm{b,ori}} - q^t_{\mathrm{b,eq}}(\boldsymbol{a}_t))^2 \end{aligned} \tag{3.4}$$

where the relevant notations are explained as follows.

First and foremost, it is expected the proposed agent-based EMADNs are physically feasible, which means that the generated EMADNs can converge in load flow calculation. This anticipation is reflected in the first term in (3.4), which imposes the penalty on the agent when producing infeasible EMADNs. $\mathbb{I}(\boldsymbol{a}_t, \boldsymbol{s}_t)$ is the infeasibility indicator function, which will become one when the load flow of the EMADN is non-convergent. P_1 is the positive penalty coefficient used for the trade-off of the multiple objectives. Besides, regarding the generated ZIP coefficients, i.e., $\boldsymbol{a}^t_{\mathrm{Z}}$ and $\boldsymbol{a}^t_{\mathrm{I}}$), if they are not feasible ($\boldsymbol{a}^t_{\mathrm{Z}}+\boldsymbol{a}^t_{\mathrm{I}} \geq \mathbf{1}$), $\mathbb{I}(\boldsymbol{a}_t, \boldsymbol{s}_t)$ will also turn to one. $\mathbb{I}(\boldsymbol{a}_t, \boldsymbol{s}_t)$ will be zero instead except for the above two scenarios. Note that function $\mathbb{I}(\boldsymbol{a}_t, \boldsymbol{s}_t)$ is non-linear, non-convex, and discontinuous because alternating current (AC) power flow equations are behind it. The characteristics of the reward design endow the formulation of EMADN problems with great flexibility.

One of the critical criteria for the fidelity of EMADNs is their capability to imitate the response of the original ADNs to the external grid, which is specified as the boundary power exchanges between the ADNs and the external grid. The third and fourth terms in (3.4) represent the quadratic power exchange errors, where $p^t_{\mathrm{b,ori}}/p^t_{\mathrm{b,eq}}$ and $q^t_{\mathrm{b,ori}}/q^t_{\mathrm{b,eq}}$ are the original/equivalent boundary active/reactive power exchange, respectively.

Another criterion for the fidelity of EMADNs is the voltage accuracy. The voltage on the retained nodes are supposed be as consistent as possible before and after network equivalencing. The second term in (3.4) represents the voltage error, where $|\cdot|$ is the absolute value operator and P_2 is the positive penalty coefficient. $V^t_{\mathrm{m,ori},i}/V^t_{\mathrm{m,eq},i}$ are the original/equivalent voltage magnitude on the retained nodes, respectively. Note that $p^t_{\mathrm{b,ori}}$, $q^t_{\mathrm{b,ori}}$, and $V^t_{\mathrm{m,ori},i}$ are assumed to be available only in the training phase.

3.4.4 Trajectory

A trajectory, also called an episode, is a sequence of states and actions. Here, the trajectory is defined as the sequence of state/action pairs in one day, where the sequence comprises multiple time steps. For example, in the subsequent case studies, the length of the sequence is set to 96 because a trajectory is set to one day, and each time interval is 15 minutes. In the training phase of the agent, the agent is supposed to experience and record different trajectories. The agent's goal is to maximize the mathematical expectation of the cumulative reward over these trajectories, which is denoted as the return of the trajectory.

In terms of the state vector defined in Fig. 3.4, $\boldsymbol{s_t}$ is comprised of the historical data in the training stage of the agent. However, in the deployment/testing stage, $\boldsymbol{s_t}$ is the data from either the forecast tool or the measurement devices. The agent will tune the parameters of EMADNs in an online manner and produce high-fidelity EMADNs that can be leveraged for computationally intensive tasks between two consecutive time steps. The EMADNs are adapted instantly once the next state is received.

3.5 Deep Reinforcement Learning

This section begins with the principles of the PPO. Two implementation details, i.e., the stochastic diagonal Gaussian policies and the early stop mechanism, are presented, followed by introducing the agent's distributed training scheme.

3.5.1 Proximal Policy Optimization

PPO is a cutting-edge DRL algorithm developed in [51]. PPO features the safe exploration of the action space and adequate exploitation of the available samples. Moreover, PPO can readily tackle the continuous and multi-dimensional action space, which is a plus for identifying the unknown parameter.

3.5.1.1 Preliminaries and Notations

PPO is an actor-critic type DRL algorithm, which comprises an actor agent π and a critic agent V. This study presents both agents by multi-layer perceptrons (MLPs) parameterized by θ and ϕ, respectively. The actor can be interpreted as the decision-making agent, in which the input is the state and the output is the action. The input to the critic is also the state, while the output is the value function, which will be defined later. Fig. 3.4 exhibits the general functions and relations of these two agents.

The advantage function, which measures how much an action is better than the others on average, is defined as:

$$A^{\pi,\gamma}\left(s_t, a_t\right) = Q^{\pi,\gamma}\left(s_t, a_t\right) - V^{\pi,\gamma}\left(s_t\right) \tag{3.5}$$

$$Q^{\pi,\gamma}\left(s_t, a_t\right) = \mathbb{E}_{\substack{s_{t+1:\infty} \\ a_{t:\infty}}}\left[\sum_{l=0}^{\infty} \gamma^l r_{t+l}\right] \tag{3.6}$$

$$V^{\pi,\gamma}\left(s_t\right) = \underset{a_t \sim \pi_\theta}{\mathbb{E}} Q^{\pi,\gamma}\left(s_t, a_t\right) \tag{3.7}$$

where γ is the discount factor; $V^{\pi,\gamma}$ and $Q^{\pi,\gamma}$ are the value function and action-value function, respectively.

3.5.1.2 PPO

To address the sample efficiency issue and stabilize the training, the PPO leverages a distinct clipped surrogate objective function based on the theory of importance sampling to perform the gradient ascent:

$$\mathbb{J}^{\pi_\theta} = \underset{s,a \sim \pi_{\theta_{\text{old}}}}{\mathbb{E}}\left[\mathbb{L}\left(s, a, \theta_{\text{old}}, \theta\right)\right] \tag{3.8}$$

$$\mathbb{L}\left(s, a, \theta_{\text{old}}, \theta\right) = \min\left(\rho_t A^{\pi_{\theta_{\text{old}}}}\left(s_t, a_t\right), \text{clip}\left(\rho_t, 1-\epsilon, 1+\epsilon\right) A^{\pi_{\theta_{\text{old}}}}\left(s_t, a_t\right)\right) \tag{3.9}$$

where ϵ is the hyperparameter that controls the permissible policy deviation; ρ_t is a ratio coefficient between the updated policy and the old policy and $\rho_t = \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}$.

The readers are referred to [51] for further analysis on (3.8). An intuitive understanding toward (3.8) and (3.9) is that the combination of *clip* and *min* functions like a regularizer on updating the π_θ, which can guarantee the safe exploration of the agent. Besides, (3.8), as the objective function of PPO, can be performed in multiple gradient steps. In contrast, the advantage actor-critic (A2C) method, the synchronous version of a popular DRL algorithm called A3C, can only perform one gradient step each epoch.

The objective function for the critic is straightforward:

$$\mathbb{J}^{V_\phi} = \mathop{\mathbb{E}}_{s,a\sim\pi_{\theta_{\text{old}}}} \left(V_\phi(s_t) - \hat{R}_t(s_t, a_t) \right)^2 \tag{3.10}$$

where the reward-to-go $\hat{R}_t := \sum_{t'=t}^{T} r_t\left(s_{t'}, a_{t'}, s_{t'+1}\right)$. Minimizing (3.10) is a regression problem, where the loss function is the mean square error. Herein the target of the output of critic is approximated by the collected reward-to-go samples. The expectation operator $\mathbb{E}$ is approximated by the mean operator in practice using the Monte Carlo (MC) approximation. For instance, $\mathbb{E}(X)$ is replaced by averaging the specific variable X over all collected samples.

3.5.1.3 Exploration via Diagonal Gaussian Policy

A policy is a rule used by an agent to decide what actions to take, and the Gaussian policy means the agent sample the actions from the Gaussian distribution. π_θ is assumed to be a multivariate Gaussian distribution, which is described by a mean vector and the diagonal covariance matrix. The outputs of π_θ are regarded as the mean of the actions $\mu_\theta(s)$. A diagonal matrix where each element is the logarithmic standard deviation $\log \sigma$ of the action in the corresponding dimension contains hyper-parameters for balancing the exploration and exploitation.

PPO trains a Gaussian policy in an on-policy way. This means that it explores by sampling actions according to its up-to-date stochastic policy:

$$\mathbf{a} = \mu_\theta(\mathbf{s}) + \sigma_\theta(\mathbf{s}) \odot \mathbf{x} \tag{3.11}$$

where $\mathbf{x}$ is the sample vector of the standard multivariate normal distribution and $\odot$ is the element-wise product.

Based on the adopted Gaussian policy assumption, $\pi_\theta(a_t|s_t)$ in (3.9) can be formulated as:

$$\pi_\theta(a_t|s_t) = \frac{1}{(2\pi)^{k/2} \prod_i^k \sigma_i} \exp(-\sum_{i=1}^{k} \frac{(x-\mu_i)^2}{2\sigma_i^2}) \tag{3.12}$$

3.5.1.4 Early Stop Mechanism

To stabilize the training further, an early stop mechanism is adopted through monitoring the Kullback-Leibler (KL) divergence, which measures the distance between probability distributions. The approximate KL divergence of outputs at time step t can be formulated as:

$$D_{\mathrm{KL}}(\pi_{\theta_{\mathrm{old},t}} \| \pi_{\theta,t}) = \log(\frac{\pi_{\theta_{\mathrm{old}}}(a_t|s_t)}{\pi_\theta(a_t|s_t)}) \tag{3.13}$$

During the process of updating the actor network within one epoch, once the average $D_{\mathrm{KL}}(\pi_{\theta_{\mathrm{old},t}} \| \pi_{\theta,t})$ over all time steps and all episodes exceeds the threshold $D_{\mathrm{KL,max}}$, the early stop mechanism will kick in to stop the following gradient ascent steps.

3.5.2 Distributed Training Scheme

This subsection focuses on implementing the training of the agent on the deep learning libraries with an auto gradient computation feature in a distributed manner.

3.5.2.1 Gradient Ascent Step

The gradient ascent step (GAS) for updating the parameters of the actor is formulated as:

$$\theta = \theta_{\mathrm{old}} + lr_\pi \nabla_\theta \mathop{\mathrm{E}}_{s,a\sim\pi_{\theta_{\mathrm{old}}}} \left[\mathbb{L}\left(s, a, \theta_{\mathrm{old}}, \theta\right)\right] \tag{3.14}$$

where ∇_θ is the gradient operator w.r.t θ and lr_π is the learning rate of the actor. GAS is dependent on the loss function formulated in (3.9), which is approximated by the MC method based on the episodic data of the current epoch.

3.5.2.2 Distributed Training

The necessity for the usage of distributed training in the EMADN problem can be explained in two aspects. First, one of the significant bottlenecks in the training efficiency of the DRL agent is that most of the time is spent on agent-environment interaction and data collection. This bottleneck is even more pronounced in the EMADN problem because each evaluation of the reward function (3.4) needs one load flow calculation with several iterations, which is obviously computationally intensive. Second, both the mini-batch training paradigm of the PPO agent and the stochastic gradient descent-type optimizer are compatible with distributed computing.

Fig.3.5. shows the proposed distributed training framework of the PPO agent. Each processor has a complete copy of the entire neural network model θ_t, whereas each processor collects episodic experiences $(\boldsymbol{s_t}, \boldsymbol{a_t})$ independently based on both samples indexes assigned to it and the actor network. Then each processor produces an exclusive loss function and gradient by performing backward error propagation locally. Finally, the processors communicate with each other to obtain an average of the gradients computed by the different processors, and the average gradient is used by GAS to update the weights θ_t.

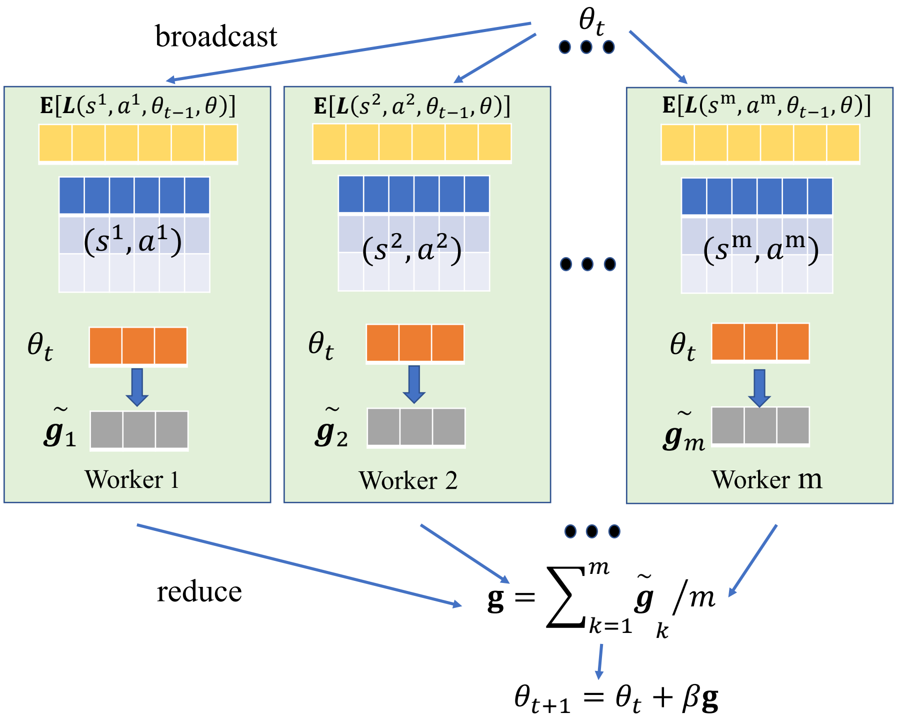

broadcast
θ_t
$\mathbf{E}[\boldsymbol{L}(s^1, a^1, \theta_{t-1}, \theta)]$
(s^1, a^1)
θ_t
$\tilde{g}_1$
Worker 1
$\mathbf{E}[\boldsymbol{L}(s^2, a^2, \theta_{t-1}, \theta)]$
(s^2, a^2)
θ_t
$\tilde{g}_2$
Worker 2
$\mathbf{E}[\boldsymbol{L}(s^m, a^m, \theta_{t-1}, \theta)]$
(s^m, a^m)
θ_t
$\tilde{g}_m$
Worker m
reduce
$\mathbf{g} = \sum_{k=1}^{m} \tilde{g}_k / m$
$\theta_{t+1} = \theta_t + \beta \mathbf{g}$

Figure 3.5: Distributed training framework of the PPO agent

Algorithm 4 summarizes the distributed training scheme of the PPO agent. Some of the relevant notations are given as follows: $\mathcal{N}_{\text{epoch}}$ is the number of training epochs; $\mathcal{N}_\pi$ / $\mathcal{N}_V$ are the number of training steps for gradient ascent/descent; m is the number of the processors; $\tilde{\mathbf{g}}_j$ represents the local gradients vector in processor j.

Algorithm 4 Distributed training scheme of the PPO agent

```
Initialize actor network $\pi_\theta$ and value function network $V_\phi$.
for $i = 0$; $i < \mathcal{N}$; $i++$ do
    Broadcast $\theta_{\text{old}}$ and $\phi_{\text{old}}$ to different processors.
    for $j = 0$; $j < m$; $j++$ do
        Policy agent $\pi_{\theta_{\text{old}}}$ interacts with the environment using (3.11) and records the trajectories samples
        $\{(s^j, a^j)\}$. Calculate the reward-to-go $\hat{R}_t^j$.
        Based on the value function $V_{\phi_{\text{old}}}$, perform the advantage estimation and obtain $A^{\pi_{\theta_{\text{old}}}}$.
        Build the loss functions of the actor and the critic based on (3.9) and (3.10). Then perform the
        gradient calculations and record the gradient $\widetilde{\mathbf{g}_j}$.
    end for
    Aggregating and averaging gradients from different processors.
    Update $\pi_\theta$ $\mathcal{N}_\pi$ times via stochastic gradient ascent based on (3.14); Update $V_\phi$ $\mathcal{N}_V$ times via stochastic
    gradient descent based on (3.10).
    $V_{\phi_{\text{old}}} \leftarrow V_\phi$; $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$.
end for
```

3.6 Numerical Result

This section covers the setups and results of the case studies on the IEEE 33-bus network. Similar case studies are also conducted using the IEEE 123-bus network to demonstrate the scalability of the proposed methods.

3.6.1 IEEE 33-bus network

3.6.1.1 System Setup

To demonstrate the effectiveness of the proposed methods, case studies are conducted on the IEEE 33-bus ADN shown in Fig.3.1. An overview of this ADN is presented in Section 3.2.1. The RESs and load data are imported from SimBench [92], which contains the whole year time series with 15-mins resolution. For each RES or load profile, there are $4*24*366 = 35,136$ data samples in total in the dataset, which is split randomly into 33,600 training samples and 1,536 test samples. Fig. 3.6 exhibits the low-carbon energy adequacy in the dataset, where the ratios of low-carbon energy production to load of training and test samples are represented in the outer and inner doughnut charts, respectively. The ratio distributions of training samples and test samples are quite close. The scenarios with more than 50% low-carbon energy penetration account for about 29% of all data samples, which is suitable to simulate the future net-zero scenario. It is assumed that there is no fossil fuels power unit inside the ADN, whereas it may import power from external grids when available RES generation is relatively low (e.g., during nights and cloudy days).

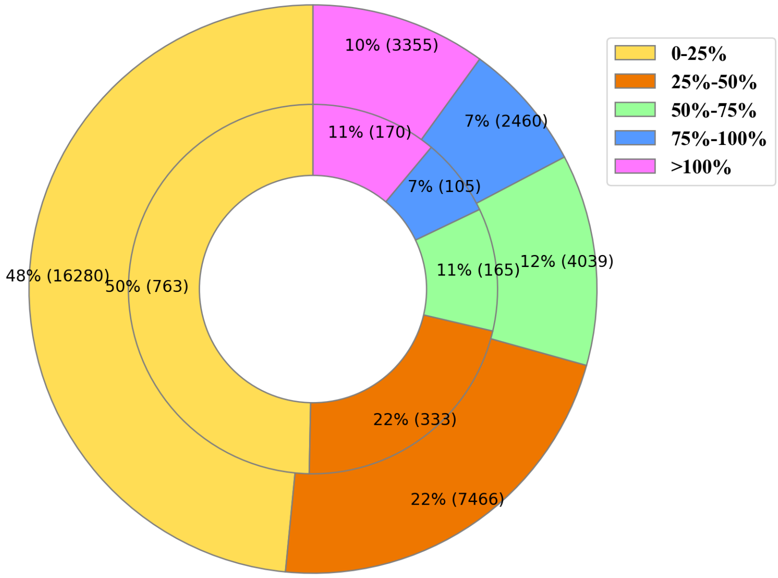


Figure 3.6: Ratio of low-carbon energy production to load.

3.6.1.2 Agent Setup

Table 3.1 shows the hyperparameters of the PPO agent. There are two hidden layers in the actor and critic MLPs, respectively. The number of neurons in each hidden layer is 64. The activation function of the hidden and output layers is the hyperbolic tangent function $Tanh$. The raw action signals are mapped into the range [a$^{\min}$,$a^{\max}$] to produce the realistic action signals using the linear transformation, where $a^{\min}$ and $a^{\max}$ restrict the parameters of the EMADNs within a reasonable range. The action dimension of the actor network in the 33-bus network is $6 * 4 + 5 * 2 + 1 + 2 = 37$. Note that it is assumed that $V_{\mathrm{m,th},3} = V_{\mathrm{m,th},2}$ in the equivalent PV plants for the simplicity of EMADNs while preserving the droop characteristic. Also, $V_{\mathrm{m,th},4}$ and $V_{\mathrm{m,th},1}$ are not considered in the equivalent PV plants that are formed through the aggregation of multiple PV plants for ensuring adequate reactive power support. Hence, the parameters of each equivalent DVRS are reduced to ξ^t

Table 3.1: Hyperparameters of the PPO agent

Parameters	Value	Parameters	Value
$\mathcal{N}$	1400	m	6
γ	0.9	ϵ	0.2
ϕ	0.95	λ	0.97
$\mathcal{N}_\pi$ / $\mathcal{N}_V$	80/80	lr_π / lr_V	3e-4/1e-3
$\log \sigma_\theta$	-0.6	$KL^{\max}$	0.015

and $\text{V}_{\text{th}}{}^t$.

The trajectory length is set to 24*4=96 steps (one day) in the case studies. Hence, there are 366 trajectories in the dataset, which are subsequently split into train and test subsets randomly. There are 350 trajectories and 16 trajectories in the train and test subsets, respectively. In the training phase, for each epoch, a mini-batch with 20 trajectories is selected randomly from the training subset to update the DRL agents. The rest protocol is highlighted in Fig. 3.4 and Section 3.4.4.

All tests are performed on a server with one NVIDIA P100 GPU and two Intel Xeon E5-2695 v4 CPUs running at 256 GB of RAM. The distributed DRL framework is implemented via Pytorch and OpenMPI.

3.6.1.3 Training Performance

The return of the trajectory collected by the agent is regarded as an index to appraise the performance of the DRL agents. Fig.3.7 shows the training curves of the DPPO and A2C agent, which depict the evolution of the average (AverageEpRet) and the standard deviation (StdEpRet) of the returns of the mini-batch trajectories. In Fig.3.7, we can observe: 1. in the first 1000 epochs, both agents exhibit excellent optimization ability. Both AverageEpRet and StdEpRet curves indicate an almost monotonic increase/decrease despite some outliers on occasion, which justifies the rationality of the formulation of the MDP and the viability of the DRL agents; 2. The DPPO agent always performs better than the A2C agent under the same training epoch, especially between epochs 100 and 600. In the last 100 epochs, the DPPO agent eventually converges to about -11.65, whereas the A2C agent converges to about -27.21. These results empirically verify that the PPO agent produces superior policy and is less likely to be trapped in the local optimum compared to the A2C agent; 3. the performance of A2C experienced two huge collapses between epoch 1000 to epoch 1100 and between epoch 1200 to epoch 1300, whereas the performance of PPO is as stable as ever. This result highlights the robustness of the PPO agent, which is one of the most prominent advantages of the PPO algorithm. The robustness of the PPO agent is attributed to two factors: 1. clipped surrogate objective function, which allows the approximately biggest safe improvement on the policy network every iteration; 2. early stop mechanism, which avoids the performance collapse resultant from the over-training. Note that the performance of the centralized version of the PPO agent (denoted as vanilla PPO) is not presented in Fig.3.7 because the distributed training mechanism has no influence on the episodic return.

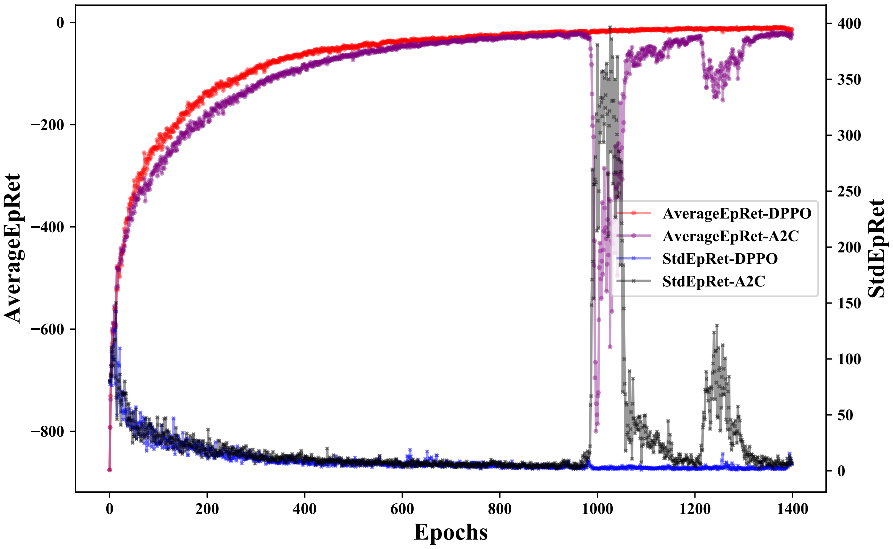


Figure 3.7: Mean and standard deviation of the episodic return of the DPPO and A2C agent over training epochs in IEEE 33-bus network.

3.6.1.4 Generalizability Performance

To verify the generalizability of the PPO agent, we use the PPO agent to build EMADNs on the test set, which contains the RESs and load scenarios the agent has never seen. Besides, the method based on nonlinear programming in [72] is employed to generate the EMADNs on the identical training and test set.

First and foremost, we find that the proposed agent-based EMADNs can converge in load flow calculations on all test samples and time steps even though these samples are inaccessible during the training. Moreover, as mentioned in Section 3.4.3, EMADNs are supposed to have the analogous response to the external grid as the original models. Fig.3.8 depicts the boundary power exchanges between the ADN and the external grid of both the equivalent and the original models. The first 96 time steps of the x-axis belong to the same day, same as every interval of 96 time steps after them. The power exchanges exhibit obvious diurnal characteristics caused by the large-scale integration of PVs. The proposed agent-based EMADNs outperform the equivalent model from [72], no matter the active or reactive power.

As for the equivalent model from [72], it can produce decent results on the active power, whereas it fails to capture the reactive power pattern of the ADNs due to the model inflexibility. In contrast to [72] which employs only a set of equivalent parameters once and for all, our method takes advantage of the DRL agents to update the parameters every time step. Another significant advantage of the proposed agent-based EMADNs is that they can maintain high fidelity, no matter whether the ADN is in the role of consumer (i.e., needs active power import) or producer (i.e., can export active power). This finding substantiates the viability of the proposed intelligent modeling framework in net-zero scenarios.

As a quantitative index, root-mean-square error (RMSE) is used to measure the distances between values from the EMADNs and the values from the original model:

$$RMSE = \sqrt{\frac{\sum_{t=1}^{T}(X^{t,\text{ori}} - X^{t,\text{eq}})^2}{T}} \tag{3.15}$$

The RMSE of the boundary power exchanges is exhibited in Table 3.2, which shows the error of our method is 51.5% less than that of [72] in active power and 84.7% less than that of [72] in reactive power.

The RMSE of the nodal voltage magnitude on the retained nodes is shown in Table 3.3, where eq and eq* represent the EMADN equipped with/without the DVRS. The results of [72] are not presented because the method in [72] cannot preserve any node. It can be

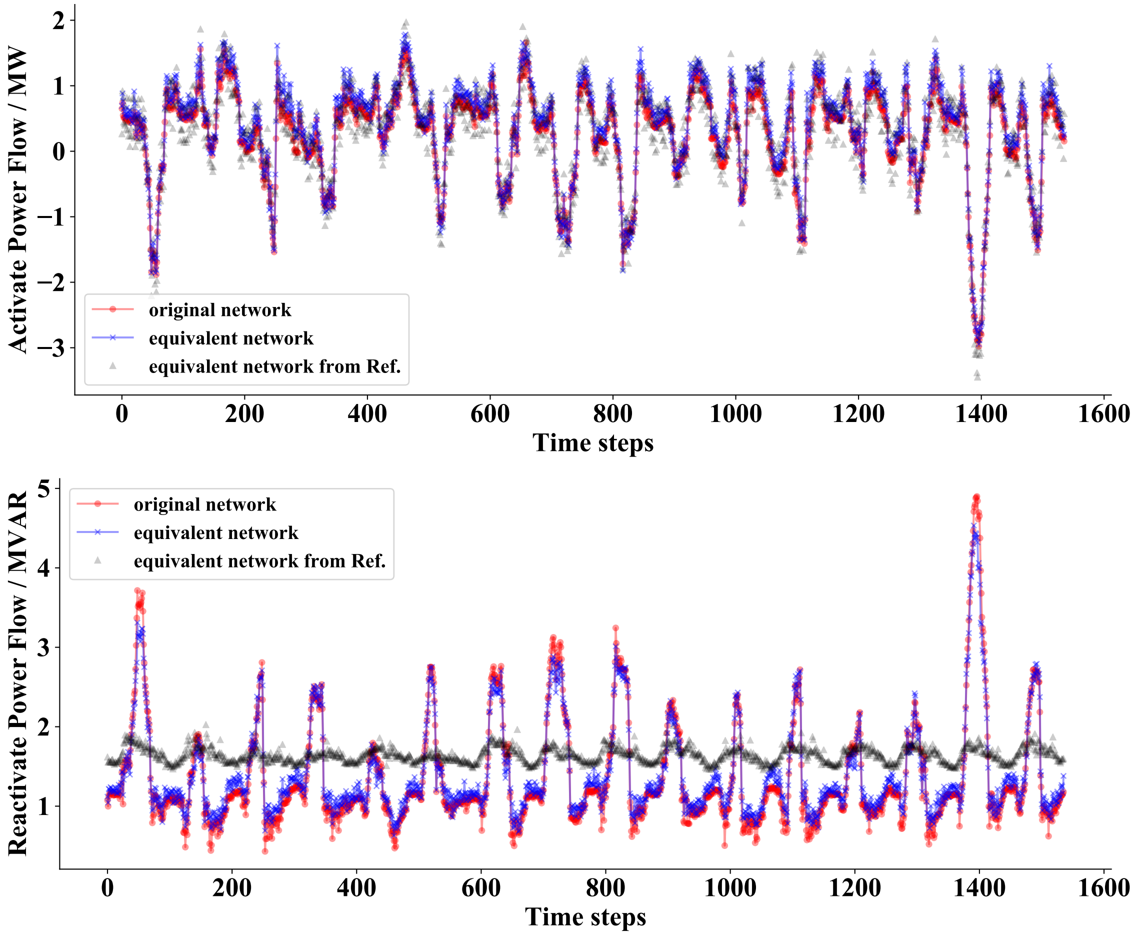


Figure 3.8: Comparison of boundary power exchanges between the original network and the equivalent networks on unseen scenarios in IEEE 33-bus network.

Table 3.2: RMSE of boundary power exchanges in IEEE 33-bus network.

RMSE / (MVA)	P	Q		P	Q	P [72]	Q [72]
Train	0.041	0.032	**Test**	0.113	0.110	0.233	0.720

Table 3.3: RMSE of nodal voltage magnitude on the retained nodes in IEEE 33-bus network.

Node-Model	2-eq	2-eq*	3-eq	3-eq*	6-eq	6-eq*
RMSE	8.4e-4	1.1e-3	1.2e-3	1.6e-3	1.7e-3	4.1e-3
Node-Model	13-eq	13-eq*	28-eq	28-eq*		
RMSE	3.3e-3	5.4e-3	2.2e-3	5.9e-3		

observed that: 1. the voltage error of the EMADNs is minor; 2. the EMADNs equipped with the DVRS have better accuracy compared to their opponent. This phenomenon is especially significant on node 6, where the equivalent PV plant is. Employing the DVRS can reduce 58% error of node 6. This result justifies the necessity of preserving the droop-based VRS of RES in EMADN; 3. The voltage error tends to increase with the distance from the root node. This is because the root node is assumed to be connected to a large power grid (transmission network) via the substation transformer, resulting in a strong voltage regulation function of the root node. Fig.3.9. compares the nodal voltage magnitude over the first 168 time steps, and similar conclusions can be drawn as mentioned above. The curves of the proposed agent-based EMADN are pretty close to the ground truth curves. These results justify that the proposed method can capture the voltage pattern of ADNs.

3.6.1.5 Discussion

In addition to voltage, users of EMADNs may expect the power flow of equivalent lines to be as close as possible to the actual line power flow. This is feasible in our proposed method. For example, if the power flows at the from-end of line 3-4 and the power flows at the to-end of line 5-6 in the 33 bus network are available in the historical data, we can augment the reward (3.4) with the difference between them and the power flows of the equivalent line 3-6. In this way, we can endow the agent-based EMADNs with accurate power flow performance.

3.6.1.6 Computation Efficiency

Table 3.4 exhibits the computation efficiency of the vanilla PPO algorithm and the distributed PPO algorithm. t^{PPO} and t^{DPPO} are the CPU time consumed by the vanilla PPO and DPPO agent running 200 epochs, respectively. We can observe that the DPPO agent is 5.3 times more efficient than the vanilla PPO. The efficiency hike is not exactly equal to the

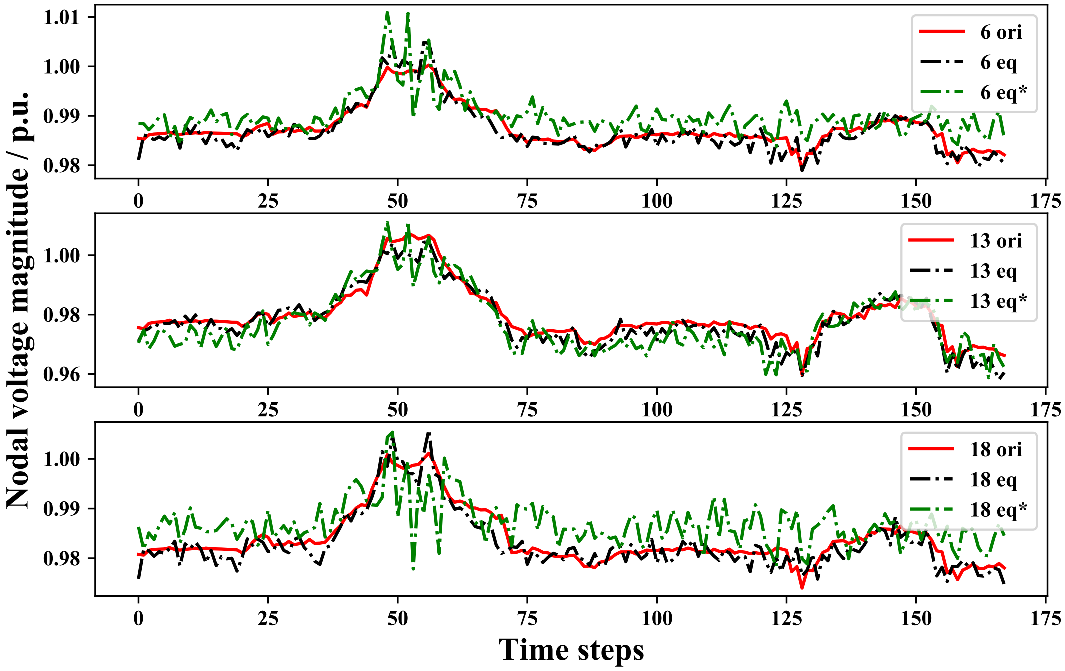


Figure 3.9: The comparison of nodal voltage magnitude on the selective retained nodes in IEEE 33-bus network.

number of workers we choose ($m = 6$) because the communication between workers takes up some time. The superiorities of the DPPO algorithm can be summarized as: 1. The relationship between training time and the number of parallel processes is approximately linear. The multi-core CPU is exploited to boost the training efficiency; 2. Since multiple workers independently explore a copy of the environment, it breaks the coupling between experiences. The benefit of this phenomenon is similar to the experience replay strategy adopted frequently in value-based DRL algorithms such as Deep Q Network. However, the DPPO avoids the excessive memory occupation of the replay buffer. In addition, it takes only 79ms for the trained agent to build an EMADN, which justifies the viability of deploying the proposed agent-based EMADN in an online manner. In addition to the privacy-preserving benefit, EMADNs show promising results in boosting the efficiency of grid analysis: it takes 205s for the EMADNs to complete the quasi-static load flow calculation of one year, whereas the original network needs 375s.

Table 3.4: Computation efficiency in IEEE 33-bus network

CPU time /(s)	t^{PPO}	t^{DPPO}		t^{exe}
Training	14,044	2,649	**Execution**	0.079

3.6.2 IEEE 123-bus network

3.6.2.1 Setup

Fig. 3.10 shows the IEEE 123-bus network and the proposed EMADN. The node indexes in Fig. 3.10 are rearranged to be consecutive using the *ext2int* function in Matpower [93] for illustration purposes. When β is selected as 5, the leaves-trimming network topological

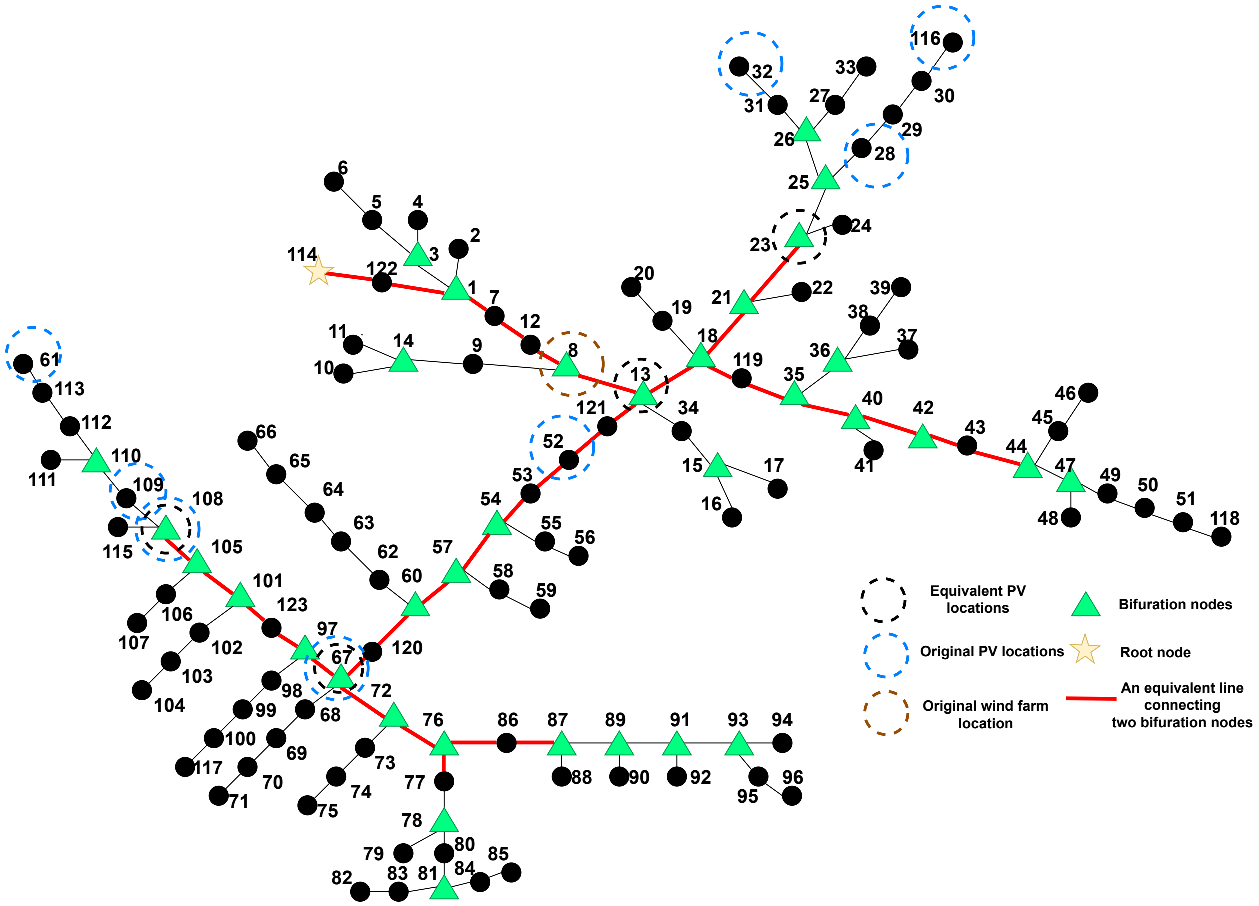


Figure 3.10: IEEE 123-bus network and the proposed equivalent model.

reduction method proposed in Section 3.3 successfully reduces the number of nodes and lines to 23 and 22, respectively, which are about 80% less than those of the original network. It is noted that only between the two remaining bifurcation points will there be an equivalent line connecting them. The only exception is the equivalent line between node 114 and node 1 because node 1 is the root node. The number of PV plants is reduced from 8 to 4 due to aggregation. Apart from the scale of the networks, the obvious distinction between the 33-bus network and the 123-bus network is the agent's action dimension. The action dimension of the actor network in the 123-bus network is $23 * 2 + 22 * 2 + 4 + 4 * 2 = 102$. Note that the ZIP load coefficients are neglected here for simplicity. Such a large action dimension causes an increase in the number of trainable parameters. π has 30,404 parameters, and V has 17,793 parameters. Furthermore, the increase in action dimension means the rise of exploration difficulty in the DRL agent. Given the enlarged challenge, the case studies in the following subsections still exhibit promising results, which demonstrates the scalability of the proposed agent-based EMADN method. Note that the scale of the EMADNs can be controlled via adjusting β in Algorithm 3.

Except for the difference in the test system, almost all setups in this subsection are the same as Section 3.6.1 except: 1. the number of neurons in each hidden layer is 128 to accommodate the larger parameter identification problem; 2. 123 different load data profiles are imported from SimBench.

3.6.2.2 Training Performance

Fig.3.11 shows the training curves of the DPPO and A2C agents in the IEEE 123-bus network. The AverageEpRet of the DPPO agent during the last 100 epochs is -34.73, whereas that of the A2C agent is -79.13. The StdEpRet of the DPPO agent during the last 100 epochs is 2.82, whereas that of the A2C agent is 5.97. These results further verify the stable and superior optimization capability of the DPPO agent relative to the A2C agent. The results can also verify that the DRL agents can handle the high dimensional and continuous action space.

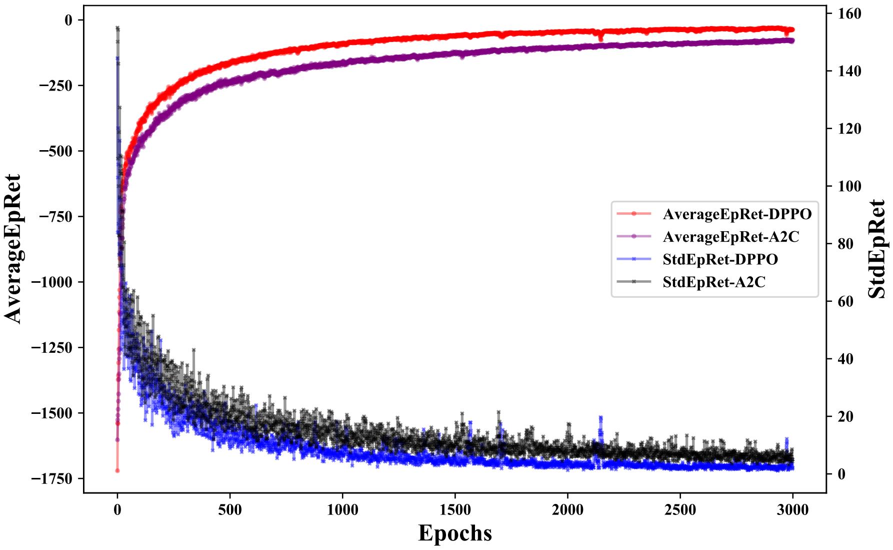


Figure 3.11: Mean and standard deviation of the episodic return of the DPPO and A2C agent over training epochs in IEEE 123-bus network.

3.6.2.3 Generalizability Performance

The generalizability of the PPO agent is evaluated on the test set in the 123-bus system. It is identified that the trained PPO agent-based EMADNs achieve 100% convergent load flow calculations. Fig.3.12 shows the boundary power exchanges results in the 123-bus system. The RMSEs for the PPO agent-based EMADNs on active and reactive power exchanges on the test set are 0.369 and 0.129, respectively, whereas those for the equivalent model from [72] are 0.399 and 0.270, respectively. Fig.3.13 shows the voltage results on the selective nodes in the 123-bus system. The overall voltage RMSEs for the EMADNs with/without the DVRS are 0.007 and 0.010, respectively. These results are aligned with those in Section 3.6.1.4, which jointly consolidates the generalizability performance of the PPO agent-based EMADNs on the larger system.

3.6.2.4 Computation Efficiency

t^{PPO} and t^{DPPO} are 4623s and 1002s for the vanilla PPO and the DPPO agent running 50 epochs, respectively. The DPPO agent is 4.6 times more efficient than the vanilla PPO in this case. Conclusions similar to those in Section 3.6.1.6 can be drawn here.

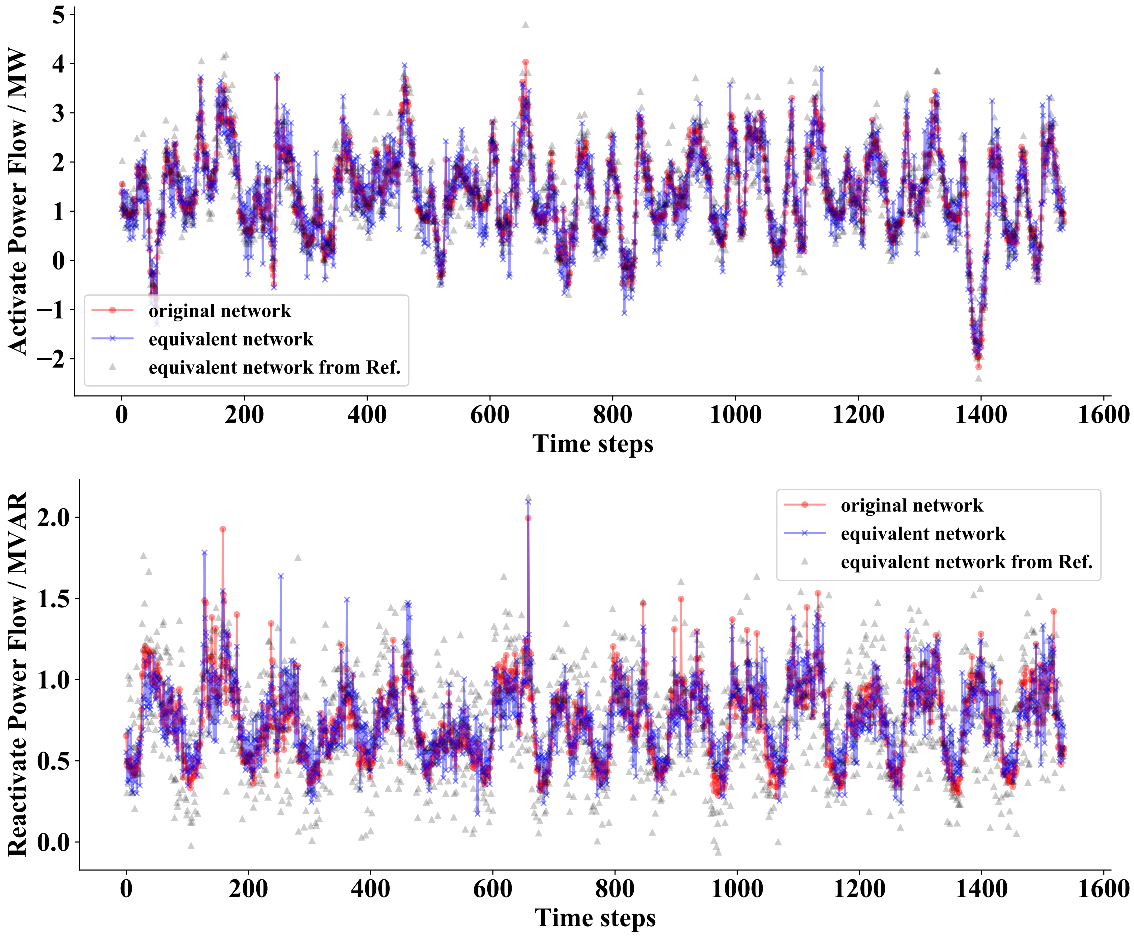


Figure 3.12: Comparison of boundary power exchanges between the original network and the equivalent networks on test set in IEEE 123-bus network.

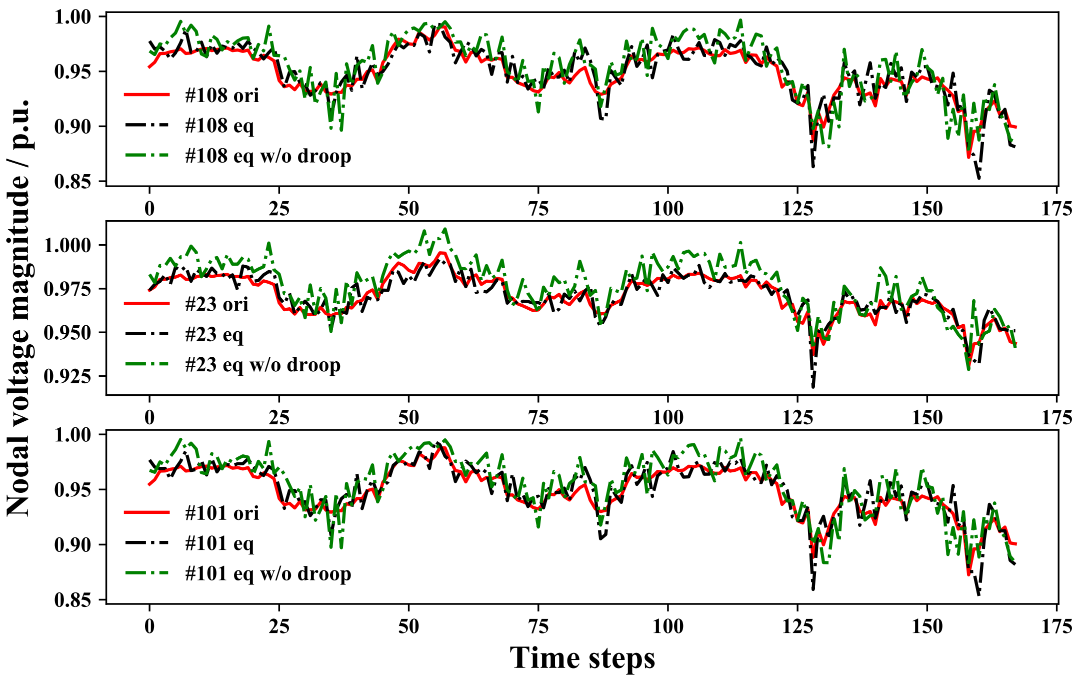


Figure 3.13: Comparison of nodal voltage magnitude on the selective nodes at IEEE 123-bus network.

3.7 Summary

This article aims at tackling the modeling challenge in the future decarbonized power sector. The uncertainty caused by the large-scale integration of low-carbon energy in the equivalent modeling of ADNs is addressed using the distributed PPO agent. A leaves-trimming network topological reduction method is proposed to retain the radial topology, the nodes of interest, and the DVRS of RESs, which is essential to the fidelity of the voltage profiles. Case studies are conducted on the IEEE-33 bus and IEEE-123 bus networks. The results exhibit the superiority of the PPO over the A2C in stability, optimization performance, and sample efficiency. The comparative results verify that the proposed EMADN can leverage the decision-making capability of the DRL and achieve the high fidelity of voltage magnitude and boundary power exchanges even in the cases of reverse power flow and unknown RESs scenarios. The comparative results between the vanilla and distributed PPO agents show the efficiency improvement brought by the parallel framework is approximately linear to the number of parallel processes. The performances in the scenarios with more than

100% renewable penetration unveil the potential of the proposed approach to facilitate the analysis of decarbonized power systems.

The authors intend to extend the proposed method to the large-scale unbalanced distribution network area in future works. The apparent hurdles lie in how to determine the phase-to-phase relationship of equivalent lines/nodes and the scalability of the DRL agent. A potential solution is using voltage sensitivity information to partition the distribution network. On the basis of multi-area partition, multi-agent DRL techniques can be developed to mitigate the scalability issue [94–97].

Chapter 4
Two-Stage Frequency-Constrained Adaptive Storage Expansion Strategy for Microgrids Using Deep Reinforcement Learning

This section addresses the strategic battery energy storage expansion planning in microgrids, challenged by the unforeseen surge in power consumption and the growth of renewable energy resources along with potential battery developments. Introducing a two-stage, multi-period planning framework, the work integrates deep reinforcement learning with linear programming, utilizing a modified rainbow algorithm with quantile regression for siting and sizing decisions. When tested on a 33-bus microgrid, the proposed approach demonstrated efficiency, scalability, and adaptability advantages over traditional mixed-integer linear programming methods and Benders decomposition.

4.1 Introduction

Recent years have witnessed the spectacular development of microgrids (MGs) as an emerging grid formation technique in the power industry. Nevertheless, the secure and economic operation of MGs is challenged by the growth of power consumption and renewable energy resources (RESs) integration, which may aggravate the peak-valley difference of net load, RES curtailment, and line congestion. Moreover, the intermittency of RESs tends to reinforce these trends, which cannot be handled with legacy MG devices and infrastructure. Energy storage is a promising remedy to the above challenges. Energy storage can contribute to the operation of MGs with great flexibility and agility. Among numerous energy storage technologies, battery energy storage (BES) is currently the most economically viable utility-scale solution [2]. Owing to the technical breakthroughs, the capital cost of BES is decreasing while the efficiency is increasing, which makes BES more economically feasible [3]. The aforementioned challenges and opportunities jointly motivate the expansion of BES in MGs for forthcoming decades. A strategic BES expansion strategy can assist decision-makers in trading off the investment risk and potential revenue.

The studies of the BES expansion strategy for power grids have aroused close attention from the industry and academia. The optimization model is a prevailing methodology for energy storage expansion. Deterministic optimization models are incapable of handling the significant uncertainty resulting from the development of RES and load in the future [98,99]. In contrast, stochastic optimization models are committed to uncertainty management, which is addressed by methods including stochastic programming (SP), robust optimization (RO), distributionally robust optimization (DRO) [100, 101], etc. Scenario-based SP methods for the BES planning in distribution systems are investigated in [102–105]. The SP methods capture the uncertain nature of extreme events and RES intermittency by scenario probability and optimize toward the minimal expected value of the objective. They are prone to providing an overly optimistic solution and need to be more scalable to massive scenario sizes. In [106], the incorporation of frequency constraints of MG islanding into MG investment planning is pioneered. The RO and a three-stage solution approach are used to manage the uncertainties. However, it ignores investments in energy storage and is not geared toward multi-year planning. Ref. [107] introduces a linearized BES sizing problem of MG, which extends the deterministic formulation to RO to address the uncertainty from hourly forecast errors. In [108], DRO is utilized to determine energy storage capacities for bulk systems, which obtains an intermediate solution between RO and SP regarding robustness. Most stochastic optimization methods end up with Mixed-Integer Linear Programming (MILP) formulation. Among these papers, [102] and [103] deal with the installation timing under the multi-year expansion context. Despite the significant advances made by stochastic optimization methods, they have been mostly restricted to uncertainties in short-time scale, e.g., intra-hour, other than the uncertainties rooted in the annual growth of RES and load. The applications of stochastic optimization methods in energy storage expansion are hindered by the formulation and computation complexities resulting from reformulation tricks and inherent stochasticity.

Recently, reinforcement learning (RL) has exhibited impressive potential for tackling numerous scenarios (e.g., online stochastic combinatorial optimization) that are entirely intractable for SP [109–111]. There is a steady stream of new developments in applying disruptive RL techniques in managing energy storage. While a majority of relevant research is carried out on the short-term energy management of the BES, few studies have attempted to investigate the application of RL in long-term planning. Ref. [112] devises an RL framework for the long-term storage expansion of MGs, where a Q-learning algorithm is used to handle the storage price variations. However, Q-learning is limited by the discrete state space and cannot tackle the continuous RES fluctuation. Ref. [113] extends the work in [112] and resorts to a deep RL (DRL) technique named double deep Q network (DDQN) to address the curse of dimensionality by neural networks (NNs) [114]. Despite the enormous progress in [112, 113], there remains a paucity of descriptions of detailed MG operation dynamics. For instance, network constraints and associated security constraints are neglected. Furthermore, storage units are typically only expected to offer backup power during power outages, and their versatility often goes untapped. The factors that restrict the direct applications of RL on storage expansion for MGs include 1) the decision-making under sophisticated operation dynamics of MGs will lead to the bottleneck due to the size of the action space and exploration efficiency; 2) it is intractable to handle massive and complicated constraints, by either simply imposing penalty terms on reward function or leveraging specific constrained RL techniques.

The focus of this work is to develop strategic and dynamic BES expansion policies that cater to both short-term operations and long-term planning of MGs, providing adaptable and flexible planning solutions for the uncertain long-term development of loads and RESs, as opposed to a fixed plan. The developed policies answer when, where, and how much BES capacity will be installed within the framework of multi-period planning. A bi-level planning framework is established, in which a sequential energy storage investment decision is made by RL for each time period in the upper level, and the lower level is a linear programming (LP) model involving the daily operation of the MG. The crafted framework effectively sidesteps the drawbacks of utilizing RL directly, delegating the decision-making of operational subproblems with numerous continuous decision variables and complex constraints to lower-level optimization algorithms. RL is only used for planning problems that involve a small number of discrete variables, which significantly avoids the bottleneck of the RL action space.

The contributions of this paper are summarized as follows:

1. In contrast to prevailing stochastic optimization-based planning approaches, this article proposes a DRL-aided bi-level multi-period BES expansion planning framework to alleviate the computational complexity and enhance planning efficiency. The proposed planning framework is a generic framework in which the decision-making for discrete and continuous decision variables on two-time scales is decoupled.

2. In the upper level, a state-of-the-art DRL agent, i.e., a rainbow algorithm with quantile regression, is leveraged to determine the installation locations and capacity sequentially. In the lower level, a tractable LP problem is formulated and solved to fulfill the optimal operation of MGs, considering the frequency constraints and backup power from BES for unexpected islanding events. As opposed to the common stochastic optimization techniques that provide one planning scheme for all uncertainties, the proposed DRL-aided approach provides dynamic planning policies, adjusting the planning solution in a rolling way to adapt to volatile future battery prices and long-term RES/load growth. The adaptivity is attributed to both the adopted Markov chain and the data augmentation technique.

3. Using the MILP-based strategies as the baselines, case studies on an MG demonstrate the agility and scalability of the DRL-aided BES expansion solution. The proposed method can provide near-optimal solutions while fulfilling four orders of magnitude efficiency improvement compared to the MILP counterparts in case studies.

The remainder of this paper is organized as follows. Section 4.3 presents an integrated planning problem benchmark with the deterministic formulation. Then, a bi-level BES planning framework is developed, where the upper-level is a combinatorial decision-making problem of siting and sizing, and the lower-level is the daily operational problem of the MG. The siting, sizing, and timing of BES installation are modeled as a Markov decision process (MDP) in Section 4.4, where Markov chain and data augmentation techniques are used to diversify the states. Afterward, a DRL with quantile regression is introduced to solve this MDP. The numerical results are exhibited in Section 4.5, followed by the concluding remarks in Section 4.6.

4.2 List of Symbols

Index and set collections:

i/k	Subscript for nodes/components
$t/d/y$	Superscript for hours/days/years
$\mathcal{P}_i/\mathcal{C}_i$	Parent node and children nodes of node i
$\mathcal{E}^{\text{NODE-NR}}$	Index set for non-root nodes
$\mathcal{E}^{\text{NODE-CAN}}$	Index set for candidate nodes for storage installation
$\mathcal{E}^{\text{dg}}/\mathcal{X}_i^{\text{dg}}$	Index set for DG units in the whole MG/at the i-th node
$\mathcal{E}^{\text{BES}}/\mathcal{X}_i^{\text{BES}}$	Index set for BES units in the whole MG/at the i-th node
$\mathcal{E}^{\text{LD}}/\mathcal{X}_i^{\text{LD}}$	Index set for loads in the whole MG/at the i-th node
$\mathcal{X}_i^{\text{RES}}/\mathcal{X}_i^{\text{ext}}$	Index set for RES and PCC at the i-th node
$\mathcal{T}/\mathcal{D}$	Set for hour/ typical and extreme day indices
$\mathcal{Y}$	Set for year indices.

Parameters:

$C^{\text{cap},y}$	Capital cost of BES unit at y-th five-year
D^d	Number of days represented by typical/extreme day d
r_{ij}/x_{ij}	Resistance/reactance of line between node i and j
V_i^{max}/V_i^{min}	Upper/lower nodal voltage magnitude limit
$\overline{p}_i^{\text{res},t,d}$	Power output forecast of RES
$p_i^{\text{D},t,d}/q_i^{\text{D},t,d}$	Nodal active and reactive power demand
$C^{\text{PCC},t}/C_i^{\text{dg},t}$	Energy price at PCC/Generation cost of DG
$C_i^{\text{e}}/C^{\text{r}}$	Operation cost of BES unit/Reserve cost of DG
$VoLL$	Value of lost load
η^{ch} / η^{dis}	Charging/discharging efficiency of BES unit
$\overline{p}_k^{\text{ch}}/\overline{p}_k^{\text{ch}}$	Charging/discharging power capacity of BES
$\overline{p}^{\text{flow}}$	Maximum power transmission on line
ϕ	Annual discount rate
H	Total inertia constant (MW·s/Hz)
η^{ch} / η^{dis}	Charging/discharging efficiency of BES unit

Variables:

$p_k^{\text{ch},t,d}/p_k^{\text{dis},t,d}$	Charging/Discharging active power of BES
$SOC_k^{t,d}/E_k^y$	SOC/energy capacity of battery
$p_k^{\text{dg},t,d}/q_k^{\text{dg},t,d}$	Active and reactive power output of DG
$v_i^{t,d}$	Squared nodal voltage magnitude
$p_i^{t,d}/q_i^{t,d}$	Nodal active/reactive power net injection
$p^{\text{ext},t,d}/q^{\text{ext},t,d}$	Active/reactive exchange power at PCC
$p_i^{\text{ls},t,d}/p_i^{\text{cur},t,d}$	Nodal load shedding/nodal RES curtailment
$R_k^{\text{U},t,d}$	Spinning-up reserve capacity of the DG unit
$p_j^{\text{flow}}/q_j^{\text{flow}}$	Active/reactive power transmission on line with end node j
$p_k^{\text{l},t}/q_k^{\text{l},t}$	Active/reactive power consumption
$p_k^{\text{res},t}$	Actual power output of RES

4.3 Bi-level multi-period BES expansion planning framework

This section first presents a MILP formulation for the deterministic BES planning problem as a benchmark. Then, stochastic factors are introduced into this deterministic problem, which is subsequently decoupled into a bi-level planning framework. Descriptions are provided for the upper and lower levels, respectively.

4.3.1 Benchmark: deterministic integrated formulation

Given specific long-term and short-term BES, RES, and load scenarios, the deterministic formulation for the BES expansion problem is given as:

$$\min J_{\text{in}} + J_{\text{op}} = \sum_{y\in\mathcal{Y}} \sum_{m\in\mathcal{E}^{\text{NODE-CAN}}} (1+\phi)^{1-y}(C^{\text{CAP},y}\overline{E}_m^y + J_{\text{op}}^y(\overline{E}_m^y)) \tag{4.1}$$

$$\text{s.t.} \quad \overline{E}_m^y \in \{0, \overline{E}_1, ..., \overline{E}_{|Z|-1}\}, \forall y \in \mathcal{Y}, \forall m \in \mathcal{E}^{\text{NODE-CAN}} \tag{4.2}$$

$$\overline{E}_i^y - z_i\overline{E}_{|Z|} \leq 0, \quad z_i \in \{0,1\}, \quad \forall i \in \mathcal{E}^{\text{NODE-CAN}} \tag{4.3}$$

$$\sum_{i\in\mathcal{E}^{\text{NODE-CAN}}} z_i \leq 1 \tag{4.4}$$

$$\text{Daily operation constraints}, \quad \forall y \in \mathcal{Y}, \forall d \in \mathcal{D}, \forall t \in \mathcal{T} \tag{4.5}$$

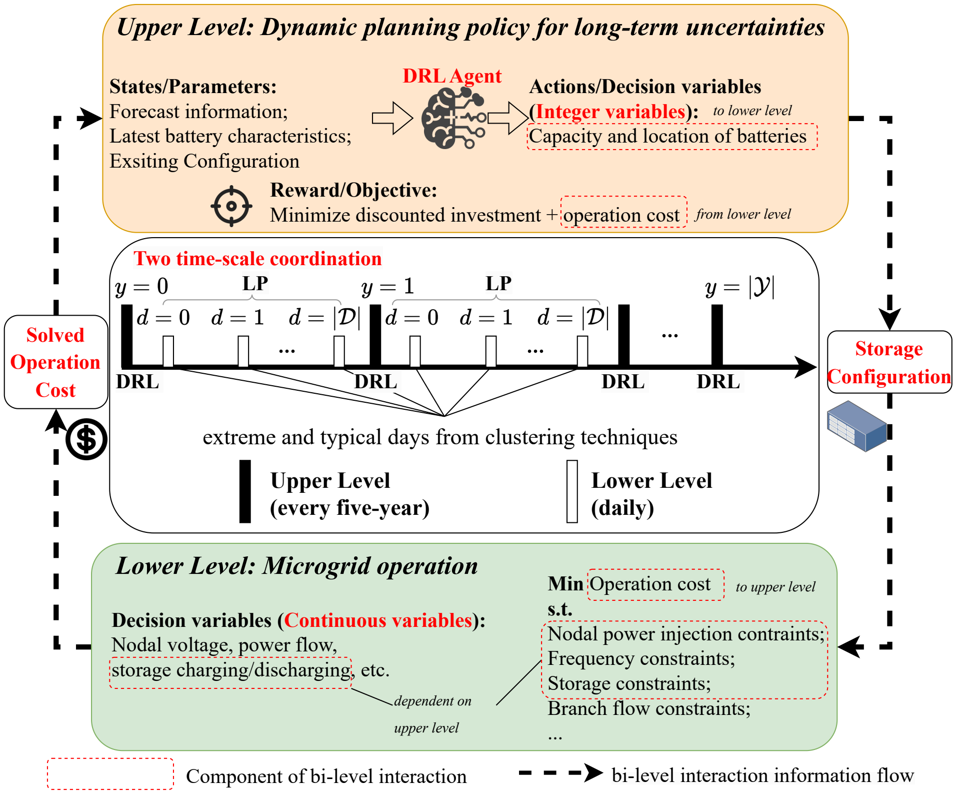

Upper Level: Dynamic planning policy for long-term uncertainties
States/Parameters:
Forecast information;
Latest battery characteristics;
Exsiting Configuration
DRL Agent
Actions/Decision variables
(Integer variables):
to lower level
Capacity and location of batteries
Reward/Objective:
Minimize discounted investment + operation cost
from lower level
Two time-scale coordination
$y = 0$
LP
$y = 1$
LP
$y = |\mathcal{Y}|$
$d = 0$ $d = 1$ $d = |\mathcal{D}|$
$d = 0$ $d = 1$ $d = |\mathcal{D}|$
...
...
...
DRL
DRL
DRL
DRL
extreme and typical days from clustering techniques
Upper Level (every five-year)
Lower Level (daily)
Solved Operation Cost
Storage Configuration
Lower Level: Microgrid operation
Decision variables (Continuous variables):
Nodal voltage, power flow,
storage charging/discharging, etc.
dependent on upper level
Min Operation cost
to upper level
s.t.
Nodal power injection contraints;
Frequency constraints;
Storage constraints;
Branch flow constraints;
...
Component of bi-level interaction
bi-level interaction information flow

Figure 4.1: DRL-aided bi-level multi-period BES expansion planning framework.

The objective function (4.1) is made up of investment cost J_{in} and operation cost J_{op}, discounted by the period discount rate ϕ. J_{in} is determined by the product of the capital cost $C^{\text{CAP},y}$ and newly installed capacity $\overline{E}_m^y$. J_{op} reflects the overall operation costs associated with (4.5) and $\overline{E}_m^y$. Eq.(4.2) is the integer constraint for $\overline{E}_m^y$, which restrict that $\overline{E}_m^y$ can only select from a finite discrete integer set Z, e.g., $\{0, 500, 1000, 2000\}$, over all available candidate nodes [115, 116]. $|*|$ stands for the cardinality of a set. Following the convention in [112], it is assumed that only one node is chosen for the installation in each planning period, which is enforced by (4.3) and (4.4). Eq.(4.5) comprises a series of linear constraints that will be detailed in Sec.4.3.3. The deterministic problem formulated in (4.1) to (4.5) is a MILP problem, in which the planning and operation problems are integrated.

4.3.2 Upper level: siting and sizing

Given the uncertainty of future BES prices, RESs, and load developments, Fig.4.1 outlines the proposed bi-level multi-period BES expansion planning framework. In the upper level, a 25-year planning horizon is subdivided into five periods of five years. The horizon and time interval here are not fixed but can be customized to adapt to different computational tasks. Adopting a 5-year interval is a common practice in multi-period or multi-stage energy storage planning studies [98, 112]. The siting and sizing decisions encoded by integer variables $\overline{E}_m^y$ are made sequentially. DRL is used to determine the siting and sizing in the upper level due to the nature of the problem, which has three key properties. Firstly, the problem is combinatorial optimization, which DRL can handle due to its capability to explore the solution space efficiently. Secondly, the problem is multi-period decision-making, which can be easily modeled as an MDP and subsequently solved by DRL algorithms. Lastly, the problem is a stochastic combinatorial optimization problem, which DRL algorithms can handle by maximizing the expected return. Supervised or semi-supervised deep learning techniques require good-quality labeled solutions, which are either unavailable or expensive to obtain in planning problems. More details on using DRL are elaborated in Sec.4.4. In Fig.4.1, the components that involve interaction between bi-levels are highlighted in red dashed boxes. The black dashed arrows explicitly show the parameters exchanged between the upper and lower levels. The upper level outputs the BES size and location of the MG as parameters for the lower level LP optimization problem. The optimization objective in the lower level, combined with BES investment cost, becomes the reward signal for the upper level DRL problem. All the required physical models are placed in the lower level, while the upper level is entirely data-driven. Our proposed framework has two stages: offline learning and online inference. In the context of this article, whether it is a model-based optimization technique or DRL-based approach, we define online decision-making as the process of providing a planning scheme given a specific set of future storage prices, renewables, and load profiles. To distinguish the offline RL learning stage from the subsequent online decision-making stage, we refer to the online decision-making stage as inference, following the convention in [117]. Optimal site selection and battery capacity decisions are only made during the online inference stage. Physical models are only needed as reward signals in offline learning. When switching to the inference stage, only the upper layer is needed, which enables a complete data-driven operation. Since the inference stage is performed in a rolling, highly efficient manner, it could be carried out online.

In contrast to the formulation in (4.1) to (4.5), the proposed bi-level framework decomposes the decision-making tasks with two different time scales: every five years and daily. The framework features 1) the decoupling of discrete and continuous variables, which enables the introduction and learning of DRL; 2) the decoupling and coordination of problems in two time scales; and 3) adaptability to stochastic scenarios, which is attributed to the use of the data-driven solution.

4.3.3 Lower level: islanding-safe operation problem

Receiving the proposed BES configuration from the upper level, the MG operation problem for one period is formulated and solved with this specific BES configuration. Specifically, we resort to an LP to minimize the summation of daily operation costs while satisfying a series of operational constraints. To alleviate the computational burden, some typical/extreme days are selected to represent the RES/load daily profiles in one period. In addition to common MG operational constraints, MG islanding-induced frequency constraints are taken into account to enforce the reliability requirements under unexpected MG islanding events.

Eq.(4.6) presents the objective function for the operation problem, which is calculated by summing up relevant costs for all typical/extreme days within a five-year period. Eq.(4.6) is weighted according to the portion of each typical/extreme day. The first term is the cost/revenue related to importing/exporting electricity via the point of common coupling (PCC) of the MG. The second term denotes the generation cost of the controllable distributed generator (DG) units, e.g., mini-size combined cycle plants, whereas the third term is the operation cost of charging/discharging storage. The fourth term represents the primary frequency response (PFR) cost of DGs, followed by the final term that sums up the penalties on the unserved load.

$$\begin{aligned}\min J_{\text{op}}^{y}(\overline{E}_{m}^{y}) = \sum_{d\in\mathcal{D}} \mathrm{D}^{d}\{\sum_{t\in\mathcal{T}}[C^{\text{PCC},t}p^{\text{ext},t,d} + \sum_{k\in\mathcal{E}^{\text{DG}}} C_{k}^{\text{dg},t}p_{k}^{\text{dg},t,d} \\ + \sum_{k\in\mathcal{E}^{\text{BES}}} C^{\text{e}}(p_{k}^{\text{dis},t,d} + p_{k}^{\text{ch},t,d}) + \sum_{k\in\mathcal{E}^{\text{DG}}} C^{\text{r}}R_{k}^{\text{U},t,d} \\ + \sum_{k\in\mathcal{E}^{\text{LD}}} VoLL\, p_{k}^{\text{ls},t,d}]\}\end{aligned} \tag{4.6}$$

It is noted that the equations below in the rest of this subsection are the daily operational constraints, of which the superscript d is omitted for the sake of brevity.

4.3.3.1 Daily Operation Constraints

The MG with radial topology is subject to a linearized branch flow model, which can be written as follows for all i in $\mathcal{E}^{\text{NODE-NR}}$ and t in $\mathcal{T}$ [118]:

$$p_{i}^{t} = \sum_{j\in\mathcal{C}_{i}} p_{j}^{\text{flow},t} - p_{i}^{\text{flow},t}, q_{i}^{t} = \sum_{j\in\mathcal{C}_{i}} q_{j}^{\text{flow},t} - q_{i}^{\text{flow},t}, \tag{4.7a}$$

$$v_{i}^{t} = v_{j}^{t} - 2\left(r_{i,j}p_{j}^{\text{flow},t} + x_{i,j}q_{j}^{\text{flow},t}\right), \quad \forall j \in \mathcal{P}_{i} \tag{4.7b}$$

where (4.7a) enforces the nodal power balance, and (4.7b) shows the relation of squared nodal voltage magnitude between neighboring nodes.

For $t \in \mathcal{T}$ and $i \in \mathcal{E}^{\text{NODE}}$, the nodal power injection is determined by:

$$\begin{aligned}\sum_{k \in \mathcal{X}_i^{\text{EXT}}} p_k^{\text{ext},t} + \sum_{k \in \mathcal{X}_i^{\text{dg}}} p_k^{\text{dg},t} + \sum_{k \in \mathcal{X}_i^{\text{RES}}} p_k^{\text{res},t} \\ + \sum_{k \in \mathcal{X}_i^{\text{BES}}} p_k^{\text{dis},t} - \sum_{k \in \mathcal{X}_i^{\text{BES}}} p_k^{\text{ch},t} - \sum_{k \in \mathcal{X}_i^{\text{LD}}} p_k^{\text{l},t} = p_i^t\end{aligned} \tag{4.8}$$

$$\sum_{k \in \mathcal{X}_i^{\text{EXT}}} q_k^{\text{ext},t} + \sum_{k \in \mathcal{X}_i^{\text{dg}}} q_k^{\text{dg},t} - \sum_{k \in \mathcal{X}_i^{\text{LD}}} q_k^{\text{l},t} = q_i^t \tag{4.9}$$

The constraints imposed on line flows for $i \in \mathcal{E}^{\text{NODE-NR}}$ and $t \in \mathcal{T}$ are:

$$-\overline{p}_i^{\text{flow}} \leq p_i^{\text{flow},t} \leq \overline{p}_i^{\text{flow}} \tag{4.10a}$$

$$-\overline{q}_i^{\text{flow}} \leq q_i^{\text{flow},t} \leq \overline{q}_i^{\text{flow}} \tag{4.10b}$$

The limits on the amount of load shedding for $i \in \mathcal{E}^{\text{NODE}}$ and $t \in \mathcal{T}$ are:

$$0 \leq p_k^{\text{ls},t} \leq p_k^{\text{D},t}, \quad \forall k \in \mathcal{X}_i^{\text{LD}} \tag{4.11a}$$

$$p_k^{\text{l},t} = p_k^{\text{D},t} - p_k^{\text{ls},t}, \quad \forall k \in \mathcal{X}_i^{\text{LD}} \tag{4.11b}$$

The limits on the amount of RES curtailment for $i \in \mathcal{E}^{\text{NODE}}$ and $t \in \mathcal{T}$ are:

$$0 \leq p_k^{\text{cur},t} \leq \overline{p}_k^{\text{res},t}, \quad \forall k \in \mathcal{X}_i^{\text{RES}} \tag{4.12a}$$

$$p_k^{\text{res},t} = \overline{p}_k^{\text{res},t} - p_k^{\text{cur},t}, \quad \forall k \in \mathcal{X}_i^{\text{RES}} \tag{4.12b}$$

The SOC evolution of BES is described by the following equations for $k \in \mathcal{E}^{\text{BES}}$:

$$SOC_k^1 = SOC_k^T = \frac{E_k}{2} \tag{4.13a}$$

$$SOC_k^t = SOC_k^{t-1} + p_k^{\text{ch},t}\eta^{\text{ch}} - \frac{p_k^{\text{dis},t}}{\eta^{\text{dis}}}, t = 2, 3, ..., T \tag{4.13b}$$

Note that the explicit constraints for averting simultaneous charging and discharging of the BES are waived because of the incorporation of cost for charging and discharging in the objective function (4.6) [119].

The limit on the amount of charging/discharging power of BES for $t \in \mathcal{T}$ and $k \in \mathcal{E}^{\text{BES}}$:

$$0 \leq p_k^{\text{dis},t} \leq \overline{p}_k^{\text{dis}} \tag{4.14a}$$

$$0 \leq p_k^{\text{ch},t} \leq \overline{p}_k^{\text{ch}} \tag{4.14b}$$

Also, for $i \in \mathcal{E}^{\text{NODE}}$, v_i should be within $[\text{V}_i^{\text{MIN}},\text{V}_i^{\text{MAX}}]$.

4.3.3.2 Frequency-induced Constraints

The disruptions that occurred outside the MG may give rise to the sudden disconnection of the MG from the external grid. The loss of the imported/exported power through the PCC caused by the unexpected islanding will lead to the instant power imbalance $p^{\text{ext},t}$. This power mismatch will jeopardize the frequency stability of the operation of the MG [120]. This study focuses on the scenario of losing imported power, resulting in the plunging of frequency as shown in the first-order swing equation [121]:

$$2H\frac{d\Delta f(t)}{dt} = \sum_{k\in\mathcal{E}^{\text{BES}}} \Delta p_k^{\text{dis}} + \sum_{k\in\mathcal{E}^{\text{DG}}} \Delta p_k^{\text{dg}} - \Delta p^{\text{ext}} \tag{4.15}$$

where the rate of change of frequency is dependent on the inertia constant H, the rapid supportive response from battery Δp_k^{dis}, the droop-based automatic additional output of controllable DG Δp_k^{dg}, and the power mismatch caused by the islanding event Δp^{ext}.

To cope with the unexpected frequency deviation, BES units can respond and output power to the full discharging capacity in milliseconds regardless of whether it is being charged or discharged [122]:

$$\Delta p_k^t = \overline{p}_k^{\text{dis}} - p_k^{\text{dis},t} + p_k^{\text{ch},t}, \forall k \in \mathcal{E}^{\text{BES}} \tag{4.16}$$

With the support of BES units, the power mismatch is mitigated to:

$$\Delta p^{\text{ext},t} = p^{\text{ext},t} - \sum_{k\in\mathcal{E}^{\text{BES}}} \Delta p_k^t \tag{4.17}$$

The response of the BES units to back up the power mismatch can be divided into three phases [122]. First, the output of BES units is boosted to the maximum to perform the inertial frequency control within an inertia response timeframe $\Delta t_{\text{inertia}}$ (e.g., 5s). Then, PFR from controllable DGs starts to take effect after DGs exceed the governor deadband, during which the BES units retain the full discharging power. This PFR timeframe is denoted as Δt_{PFR}. The frequency nadir will be reached during the PFR timeframe. A common value for Δt_{PFR} is 25s. Finally, the secondary frequency response will be activated to bring the frequency back to the normal range. This timeframe is denoted as Δt_{SFR}, and its typical value is 300s. During t_{SFR}, the discharging of BES units is decreased linearly to zero.

The potential SOC variance during the above three phases can be derived by:

$$\Delta SOC_k = \overline{p}_k^{\text{dis}} \left(\Delta t_{\text{inertia}} + \Delta t_{\text{PFR}} + 0.5 \Delta t_{\text{SFR}}\right) \tag{4.18}$$

The operation of the BES should ensure the available SOC can satisfy the potential SOC decrease:

$$SOC_k^t \geq \Delta SOC_k, \forall t \in \mathcal{T}, \forall k \in \mathcal{E}^{\text{BES}} \tag{4.19}$$

An essential metric for the frequency transient process during the MG islanding ride-through is the rate of change of the frequency. The post-contingency initial rate of change of the frequency $RoCoF$ should be maintained under the maximal limit $\overline{RoCoF}$ to avoid triggering relay [120]:

$$RoCoF = \frac{d\Delta f(t)}{dt} = \frac{\Delta p^{\text{ext},t}}{2H} \leq \overline{RoCoF} \tag{4.20}$$

The frequency nadir is the other vital frequency metric. It corresponds to the lowest point of the frequency dynamical process. Operators of the MG have an indicator for the lowest acceptable frequency $f^{\min}$. The violation of this indicator will destabilize the MG, leading to under-frequency load-shedding and even large-scale frequency collapse.

The timely delivery of the PFR from the controllable DGs to address the power mismatch enforces:

$$R_i^{\text{U},t} \Delta p^{\text{ext},t} \leq 2\lambda_i H \left(f^0 - f^{\min} - f^{db}\right), \forall i \in \mathcal{E}^{\text{DG}} \tag{4.21}$$

where λ_i is the inherent ramping rate for the governor of i-th DG. f^0 and f^{db} are the nominal frequency and the dead band frequency of generators, respectively.

An adequate primary reserve is needed to cover the power mismatch, and it is a necessary condition for satisfying the frequency nadir requirement:

$$\sum_{k \in \mathcal{E}^{\mathrm{DG}}} R_k^{\mathrm{U},t} \geq \Delta p^{\mathrm{ext},t} \tag{4.22}$$

Eqs.(4.21) and (4.22) make up a sufficient condition for maintaining frequency nadir above the predefined $f^{\min}$ [121].

The limits on the primary reserve of the DGs are:

$$0 \leq R_k^{\mathrm{U},t} \leq \overline{p}_k^{\mathrm{dg}} - p_k^{\mathrm{dg},t}, \quad \forall k \in \mathcal{E}^{\mathrm{DG}}, \forall t \in \mathcal{T} \tag{4.23}$$

The only nonlinear constraint among Eqs. (4.7) to (4.23) is (4.21), which contains a bilinear term. A McCormick envelope can replace this bilinear term to get a piecewise linear relaxation. Interested readers are referred to the appendix of [122] for details of the reformulation technique. Introducing auxiliary variables $W_{i,k}^{\mathrm{dis},t} = R_i^{\mathrm{U},t} p_k^{\mathrm{dis},t}$ and $W_{i,k}^{\mathrm{ch},t} = R_i^{\mathrm{U},t} p_k^{\mathrm{ch},t}$, (4.21) is substituted by:

$$W_{i,k}^{\mathrm{ch},t} - \overline{p}_i^{\mathrm{dg}} p_k^{\mathrm{ch},t} - \overline{p}_k^{\mathrm{ch}} R_i^{\mathrm{U},t} + \overline{p}_i^{\mathrm{dg}} \overline{p}_k^{\mathrm{ch}} \geq 0 \tag{4.24a}$$

$$W_{i,k}^{\mathrm{ch},t} \leq \overline{p}_k^{\mathrm{ch}} R_i^{\mathrm{U},t}, \quad W_{i,k}^{\mathrm{ch},t} \leq \overline{p}_i^{\mathrm{dg}} p_k^{\mathrm{ch},t} \tag{4.24b}$$

$$W_{i,k}^{\mathrm{dis},t} - \overline{p}_i^{\mathrm{dg}} p_k^{\mathrm{dis},t} - \overline{p}_k^{\mathrm{dis}} R_i^{\mathrm{U},t} + \overline{p}_i^{\mathrm{dg}} \overline{p}_k^{\mathrm{dis}} \geq 0 \tag{4.24c}$$

$$W_{i,k}^{\mathrm{dis},t} \leq \overline{p}_k^{\mathrm{dis}} R_i^{\mathrm{U},t}, \quad W_{i,k}^{\mathrm{dis},t} \leq \overline{p}_i^{\mathrm{dg}} p_k^{\mathrm{dis},t} \tag{4.24d}$$

4.4 Markov decision processes and DRL

This section designs a specific MDP for the siting, sizing, and timing problem in the upper-level planning problem. Markov chain and data augmentation (DA) techniques are also introduced to capture future uncertainties and enrich the variance of state vectors. A DRL algorithm based on quantile regression is then introduced to learn the optimal planning policy for the designed MDP.

4.4.1 Markov decision processes

MDP is specified by a 5-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}(s_{y+1} = s' \mid s_y = s, a_y = a), R(s,a), \gamma)$, where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{P}$ is the state transition probability, $R(s,a)$ is the reward function, and γ is the reward discount factor. The design rationale for each component is clarified below.

4.4.1.1 State space

States are resources and information needed for decision-making. The state space is supposed to reflect the current operating conditions and projections for the next upcoming period. As such, the state vector is defined as:

$$\begin{aligned}\mathbf{s_y} := \big(& C^{\text{CAP},y}, \eta^{\text{ch},y}, \eta^{\text{dis},y}, \psi^{\text{res},y}_{\text{mean}}, \psi^{\text{res},y}_{\text{max}}, \psi^{\text{res},y}_{\text{min}}, \\ & \psi^{\text{l},y}_{\text{mean}}, \psi^{\text{l},y}_{\text{max}}, \psi^{\text{l},y}_{\text{min}}, \sum_{n=1}^{y-1} \sum_{m \in \mathcal{E}^{\text{NODE-CAN}}} \overline{E}^{n}_{m}, \mathbf{\Pi}^{y})\end{aligned} \tag{4.25}$$

where $\psi^{\text{res},y}_{\text{mean}}$, $\psi^{\text{res},y}_{\text{max}}$, and $\psi^{\text{res},y}_{\text{min}}$ are the mean/maximum/minimum normalized RES capacity projections for the y-th five-year period, respectively. Similarly, $\psi^{\text{l},y}_{\text{mean}}$, $\psi^{\text{l},y}_{\text{max}}$, and $\psi^{\text{l},y}_{\text{min}}$ are the mean/maximum/minimum normalized load ratio projections for the y-th five-year period, respectively. $\sum_{n=1}^{y-1} \sum_{m \in \mathcal{E}^{\text{NODE-CAN}}} \overline{E}^{n}_{m}$ denotes the storage capacity that has been installed in the MG by the y-th five-year period. $\mathbf{\Pi}^{y}$ is an installed location indicating vector with each element corresponding to one candidate location. The element in $\mathbf{\Pi}^{y}$ is one if at least one BES has been installed at the corresponding location, and zero otherwise.

4.4.1.2 Action space

The DRL agent decides the action (i.e., siting and sizing scheme for the current five-year period) to take upon procuring the state vector in (4.25). The time scale for each action to be taken is five-year. As the planning horizon is 25 years, there are five actions and time steps in one episode.

Consistent with Eq.(4.2), the discrete action space is leveraged in this study, and its dimension is set to $N_{\text{act}} = |Z| \times |\mathcal{E}^{\text{NODE-CAN}}|$. The design of the action space with discrete capacity levels aims to avoid the introduction of hybrid action spaces and to consider the actual minimum capacity of the wholesale battery pack. The number of capacity levels, the capacity range, and the number of candidate nodes should be determined based on the specific system and parameters to strike an equilibrium between efficient exploration of the DRL agent and model granularity.

An arbitrary point n in this discrete action space can be mapped to the siting and sizing decision by:

$$\begin{aligned} i = n \bmod (Z), j = n \operatorname{div} (Z), \\ \overline{E}_j^y = \overline{E}_i, \overline{E}_m^y = 0, \forall m \in \mathcal{E}^{\text{NODE-CAN}}, m \neq j \end{aligned} \tag{4.26}$$

where mod and div are the remainder and quotient operators, respectively. For example, suppose there are six candidate locations ($\{2, 7, 21, 24, 26, 30\}$) and four candidate capacities ($\{0, 500, 1000, 2000\}$), then $N_{\text{act}} = 24$. If the action outputted by the DRL agent is 18, then $i = 2$ and $j = 4$. According to the transformation in Eq. (4.26), we can obtain the decision to install 1000kWh at node 26.

4.4.1.3 State transition probability and data augmentation

State transition probability captures all system dynamics that govern the environment, namely the long-term planning problem. As the expansion project spans nearly 25 years in this study, a stationary or constant growth rate assumption on the variations of the future storage price, RES capacity, and load amount employed in [102] is no longer valid. In terms of future storage price, it is more practical to describe the downtrend using a discrete-time Markov chain (DTMC). DTMC is a discrete event sequence in which the occurrence of the event at the next time step depends only on the current state. The transition of the events in a DTMC is characterized by transition probability. Fig.4.2 exhibits a demonstrative DTMC for future battery storage price based on the data from [3]. The three states in each layer except in layer 0 represent the low, middle, and high projections for the capital costs of BES units, respectively. The numeric values on the unidirectional edges between nodes are the transition probabilities between storage prices. This DTMC characterizes the probability transition of C^{cap} in the state vector. During an episode, the collection of C^{cap} is a sample of this DTMC.

The calendar aging of the installed BES units is described by introducing an annual 1% decline in energy capacity. The transition of the charging and discharging efficiency is

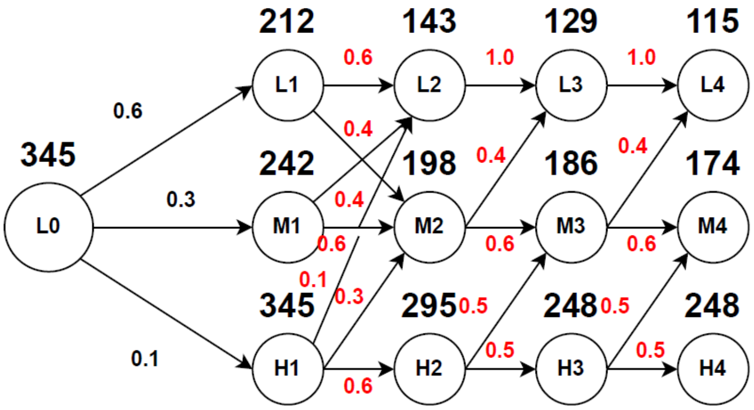


Figure 4.2: A demonstrative DTMC for battery storage price.

assumed to be deterministic and an uptrend to reflect future technological advancement. For instance, the transition trajectories for η^{ch} and η^{dis} are set to $[0.9, 0.92, 0.94, 0.95, 0.96]$ and $[0.92, 0.94, 0.96, 0.97, 0.98]$, respectively, in the case studies.

The transition of states associated with installed capacity and installed locations is dependent completely on the previous decisions made by the DRL agent.

Describing the evolution of the load and RES profiles in the coming decades by one or multiple curves (or trajectories) are not adequate enough. This study leverages the DA technique to generate abundant and diversified load and RES profiles, which are believed to be capable of fully representing potential development scenarios. Another reason for the adoption of DA is that it is essential to the generalization ability of the DRL agent. It is of paramount importance to have a DRL agent that is adaptive to uncertainty and can produce near-optimal planning decisions simultaneously.

The DA for future RES and load development is implemented via Algorithm 5, which can output a stochastic monotonically increasing list to compose a complete RES/load increase trajectory. The core of Algorithm 5 is that the conditional probability distribution of o_{i+1} given the value of o_i is a uniform distribution $\mathcal{U}_{[\underline{O}, o_i]}$, where o_i is also a random variable that follows a uniform distribution. An example of a RES evolution trajectory is demonstrated as $L^{\text{res}} = [1, 1.1, 1.2, 1.25, 1.42]$, where the installed capacity of RES in the third five-year period is 20% higher than that in the first five-year period. Based on the generated trajectory L, a complete set of profiles for the next 25 years are produced by multiplying the RES and load profiles from a benchmark year with each growth rate from L. To further diversify the learning data and account for the intra-day uncertainty of RES and load, these artificial profiles are perturbed by adding Gaussian noises derived from a standard Gaussian distribution and re-scaled by the magnitude of the original value. The $\overline{\text{O}}$, $\underline{\text{O}}$, and the profiles of the benchmark year jointly determine the future profiles considered in the planning horizon. Users can tune them to accommodate the needs of different systems and scenarios. In the learning phase, the statistical quantities related to RES and load in the state vector are obtained based on those artificial future profiles. In the inference phase, these statistical quantities can be obtained through the latest forecasting data.

Algorithm 5 Increasing sequence generator

```
Input:
    $|\mathcal{Y}|$, $\overline{\text{O}}$ maximum increase ratio, $\underline{\text{O}}$ minimum increase ratio
Output:
    Monotonically increasing list $L$ with $|\mathcal{Y}|$ elements
Initialize an empty list $L$, $U = \overline{\text{O}}$, $B = \underline{\text{O}}$
for each $i \in [0, |\mathcal{Y}|]$ do
    $o_i = \mathcal{U}_{[B,U]}$.sample(), where $\mathcal{U}_{[B,U]}$ is a continuous uniform distribution with minimum and maximum parameters $B$ and $U$.
    $L$.append($\mathrm{o_i}$)
    $U = o_i$
end for
Reverse the ordering of $L$
```

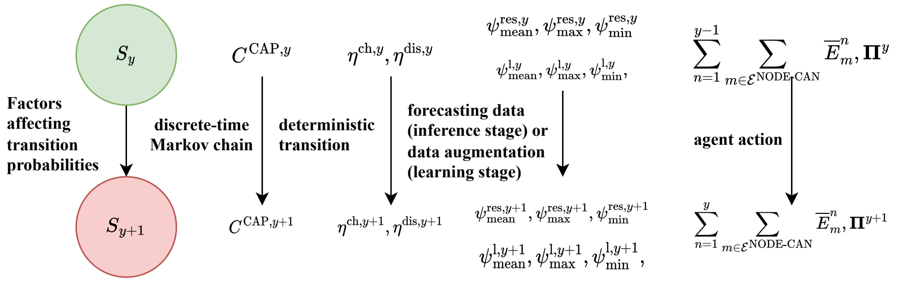

S_y
S_{y+1}
Factors affecting transition probabilities
$C^{\mathrm{CAP},y}$
$C^{\mathrm{CAP},y+1}$
discrete-time Markov chain
$\eta^{\mathrm{ch},y}, \eta^{\mathrm{dis},y}$
$\eta^{\mathrm{ch},y+1}, \eta^{\mathrm{dis},y+1}$
deterministic transition
$\psi^{\mathrm{res},y}_{\mathrm{mean}}, \psi^{\mathrm{res},y}_{\max}, \psi^{\mathrm{res},y}_{\min}$
$\psi^{\mathrm{l},y}_{\mathrm{mean}}, \psi^{\mathrm{l},y}_{\max}, \psi^{\mathrm{l},y}_{\min},$
forecasting data (inference stage) or data augmentation (learning stage)
$\psi^{\mathrm{res},y+1}_{\mathrm{mean}}, \psi^{\mathrm{res},y+1}_{\max}, \psi^{\mathrm{res},y+1}_{\min}$
$\psi^{\mathrm{l},y+1}_{\mathrm{mean}}, \psi^{\mathrm{l},y+1}_{\max}, \psi^{\mathrm{l},y+1}_{\min},$
$\sum_{n=1}^{y-1} \sum_{m\in\mathcal{E}^{\text{NODE-CAN}}} \overline{E}^{n}_{m}, \mathbf{\Pi}^{y}$
agent action
$\sum_{n=1}^{y} \sum_{m\in\mathcal{E}^{\text{NODE-CAN}}} \overline{E}^{n}_{m}, \mathbf{\Pi}^{y+1}$

Figure 4.3: Factors affecting state transition.

4.4.1.4 Reward function and reward discount factor

The design of the reward function is straightforward. It comprises the investment cost and the subsequent operation of the current period, weighted by ϕ:

$$r_y = -(1+\phi)^{1-y}\left(J^{y}_{\mathrm{op}} + \sum_{m\in\mathcal{E}^{\text{NODE-CAN}}} C^{\mathrm{CAP},y}\overline{E}^{y}_{m}\right) \tag{4.27}$$

The reward discount factor enables DRL agents to adjust their behavior, focusing on either long-term or short-term goals. The reward discount factor is set as a scalar γ smaller than one in this study.

4.4.2 Distributional DRL

Rainbow algorithm is a state-of-the-art DRL algorithm proposed by Deepmind for a discrete action space [123]. It synergistically combines five technical extensions along with the DDQN: prioritized experience replay (PER), dueling network, noisy network, distributional DRL, and N-step learning. To accelerate and stabilize the learning of RL algorithms, the N-step learning technique uses multiple rewards for learning. More details on these extensions can be found in [123]. This work advances the vanilla rainbow algorithm by incorporating the quantile regression DQN (QR-DQN) technique.

4.4.2.1 Double deep Q network

Building on the Q-learning algorithm that relies on the optimal action-value function $Q^{\star}$ to select the optimal action, deep Q network (DQN) approximates the optimal action-value function using NN $Q\left(s, a; \mathbf{w}\right)$ parameterized by $\mathbf{w}$. Temporal difference (TD) learning is employed to train $Q\left(s, a; \mathbf{w}\right)$ in the DQN.

Per TD learning, TD error δ_t constitutes the loss function to the DQN: $\delta_y = Q\left(s_y, a_y; \mathbf{w}\right) - \zeta_y$, where ζ_y is denoted as the TD target. DDQN creates a target network $Q\left(s, a; \mathbf{w}^{-}\right)$ to evaluate the TD target $\zeta_y := r_y + \gamma \cdot Q\left(s_{y+1}, a^{\star}; \mathbf{w}^{-}\right)$, where $a^{\star}$ denotes the optimal action for next state. The selection of $a^{\star}$ is according to DQN: $a^{\star} = \underset{a}{\mathrm{argmax}} Q\left(s_{y+1}, a; \mathbf{w}\right)$.

4.4.2.2 Distributional DRL with Quantile Regression

As opposed to regular DRL algorithms that learn the Monte Carlo approximation of the mean of the optimal value function, distributional DRL algorithms learn the underlying distribution. It is believed that capturing the landscape of the values can improve the performance of the DRL agent [123]. This work replaces the categorical DQN adopted in [123] with QR-DQN proposed in [124], which outperforms the categorical DQN because it is not bounded to a specific range of values and thus can adapt to a broader range of states.

QR-DQN learns a parameterized distribution on fixed quantiles. If the number of quantiles is specified as N, the probability for each quantile is $\frac{1}{N}$. The quantile vector is denoted as $\hat{\tau}$, in which each element is a quantile denoted by $\hat{\tau}_i$ and $\hat{\tau}_i \in [0, 1]$. An example of $\hat{\tau}$ is $[0.25, 0.50, 0.75, 1]$ when $N = 4$. To set the outputs of the QR-DQN to the quantile values corresponding to each action, the output layer of Q network is expanded to $N \times \mathcal{A}$, which means every N neurons corresponds to one action:

$$Q(s, a) := \mathbb{E}[\theta(s, a)] = \frac{1}{N} \sum_{i=1}^{N} \theta_i(s, a) \tag{4.28}$$

where θ_i is i-th support, i.e., the i-th quantile value for one action value.

A quantile Huber loss function $\rho_{\hat{\tau}}^{\kappa}$ for TD error δ is designed to update the NN:

$$\rho_{\hat{\tau}}^{\kappa}(\delta) = \left|\hat{\tau} - \Upsilon_{\{\delta < 0\}}\right| \mathcal{L}_{\kappa}(\delta) \tag{4.29}$$

$$\mathcal{L}_{\kappa}(\delta) = \begin{cases} \frac{1}{2}\delta^2, & \text{if } |\delta| \leq \kappa \\ \kappa\left(|\delta| - \frac{1}{2}\kappa\right), & \text{otherwise} \end{cases} \tag{4.30}$$

$\mathcal{L}_\kappa(\delta)$ represents a Huber loss. $\Upsilon_{\{\delta<0\}}$ denotes an indicator vector, where its elements are 1 when $\delta < 0$ and 0 otherwise. $\left|\hat{\tau} - \Upsilon_{\{\delta<0\}}\right|$ imposes an asymmetric adjustment to $\mathcal{L}_\kappa(\delta)$, penalizing overestimated δ with weight $1 - \tau$ and underestimated δ with weight τ. κ is a hyperparameter and is set to 1 in this study.

To derive the TD error δ in QR-DQN, the target action value distribution is approximated by using the Bellman optimality operator, which yields:

$$\begin{aligned}\mathcal{T}\theta(s,a) &:= r(s_y, a_y) \\ &\quad + \gamma\theta\left(s_{y+1}, \arg\max_{a'} Q\left(s_{y+1}, a'; \mathbf{w}\right); \mathbf{w}^-\right)\end{aligned} \tag{4.31}$$

where $\mathcal{T}$ denotes TD target operator. Note that the DDQN is integrated here by using different NNs for action selection and target distribution evaluation.

Combining (4.28) to (4.31) yields the final loss function:

$$\begin{aligned}\mathcal{L}_{QR}^{\hat{\tau}}(\theta) &:= \sum_{i=1}^{N} \mathbb{E}_j\left[\rho_{\hat{\tau}_i}^{\kappa}\left(\mathcal{T}\theta_j - \theta_i\right)\right] \\ &= \sum_{i=1}^{N} \frac{1}{N} \sum_{j=1}^{N} \rho_{\hat{\tau}_i}^{\kappa}\left(\mathcal{T}\theta_j - \theta_i\right)\end{aligned} \tag{4.32}$$

4.4.2.3 Implementation

Algorithm 6 outlines the implementation of the proposed bi-level BES expansion planning framework using the Rainbow algorithm with quantile regression (Rainbow-QR). Specifically, steps 7, 8, and 15 in Algorithm 6 explicitly explain the information exchanged between the upper and lower levels.

4.5 Numerical Results

This section is dedicated to the case studies on a 33-bus radial MG to demonstrate the effectiveness of the proposed bi-level BES planning framework. Test configuration, comparative experiments, scheme analysis, and generalizability verification are presented, respectively.

Algorithm 6 Rainbow-QR algorithm for the bi-level BES expansion planning

```
 1: Initialize original and target QR-DQNs $\mathbf{w}$ and $\mathbf{w}^{-}$; $y := 0$
 2: for $i \in [0, N_{\text{learn}}]$ do
 3:     if $y == 0$ then
 4:         Sample a BES price trajectory from the DTMC in Fig.4.2.
 5:         Generate a RES/load increase trajectory by Algorithm 5.
 6:     end if
 7:     Upper level: Form $s_y$ as in (4.25) and select action according to $a_y = \arg\max_{a'} Q\left(s_y, a'; \mathbf{w}\right)$. The BES capacity and site decoded from $a_y$ are sent to the lower level.
 8:     Lower level: Using the existing storage configuration as input parameters, solve the LP problem formulated in (4.6) to (4.20), and (4.22) to (4.24d) to obtain the optimal operation cost, which is sent back to the upper level to compose the reward function $r_y$.
 9:     Form $s_{y+1}$ with $a_y$ and trajectories from steps 4 and 5.
10:     Store a transition $(s_y, a_y, r_y, s_{y+1})$ in the buffer.
11:     $y \leftarrow y + 1$.
12:     if $y == |\mathcal{Y}|$ then
13:         $y \leftarrow 0$.
14:     end if
15:     Upper level: Evaluate the QR loss function in (4.32) with the reward from the lower level. Update the planning policy $\mathbf{w}$ with an optimizer.
16:     Update target QR-DQN $\mathbf{w}^{-} \leftarrow \mathbf{w}$ every $N_{\text{update}}$
17: end for
```

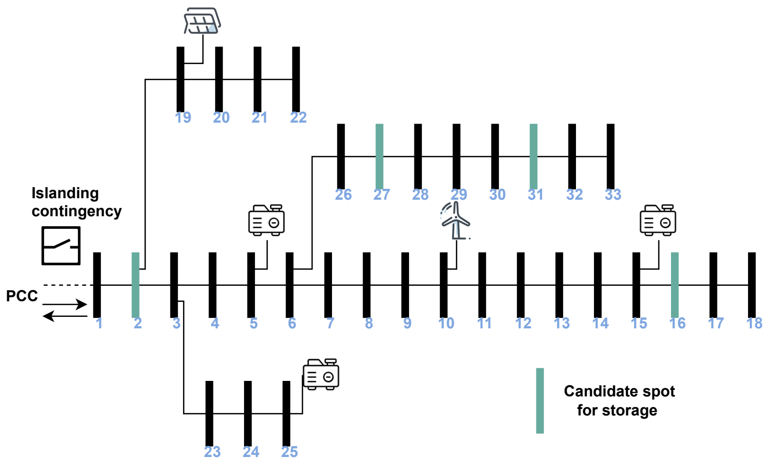


Figure 4.4: IEEE 33-bus MG with candidate spots for storage, controllable DGs, and renewable generation.

4.5.1 Test configuration

Fig.4.4 shows the layout and components of the MG modified from the IEEE 33-bus system, which integrates three dispatchable DGs, one PV cluster, and one wind cluster. The baseline load is 3.7MW/2.3MVar, and the initial RES penetration ratio is set as 30%. The network parameters, DRL hyperparameters, and optimization-related parameters are provided in [125].

The BES projection data comes from [3], and is shown in Fig.4.2. We collect hourly RES and load profiles over a year from PJM and National Renewable Energy Laboratory, respectively. These data are integrated to serve as the benchmark data, and they are made publicly available on [125]. The stochastic multi-period growth and short-term variation are added to the benchmark data to create the learning data for the DRL agents. Fig.4.5 shows a example of some learning data.

To reduce the computational complexity of the optimal operation problem, typical and extreme days are selected and displayed in Table 4.2. First, the optimal operation problem based on the baseline data is solved to calculate the daily cost for each day. Statistical analysis of quartiles is performed on the calculated daily costs to select extreme days. After excluding these extreme days, a k-medoids method is employed to pick typical days based

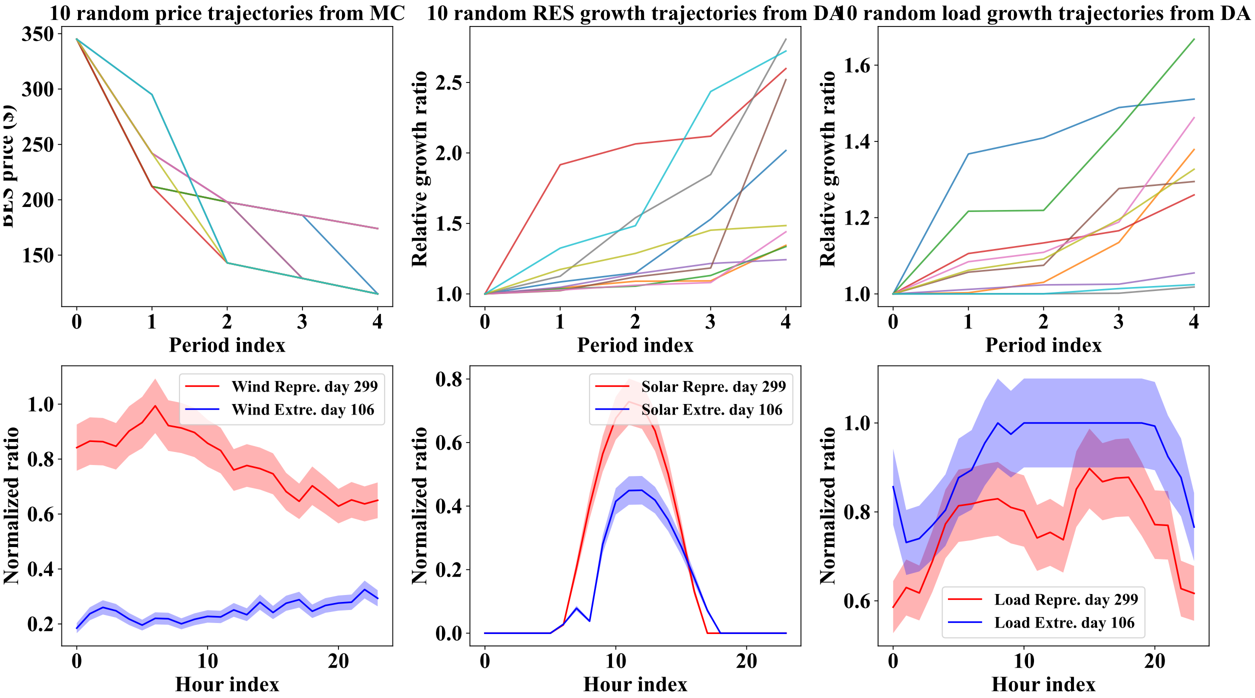


Figure 4.5: Upper: period-level price, RES, and load trajectories; Lower: examples of hourly wind, solar, and load profiles with variation intervals

Table 4.2: Typical and extreme days.

Typical days (day index:counts)	(299:135, 186:108, 118:45, 95:40, 101:18, 143:4)
Extreme days (day index:counts)	(106:1, 122:1, 127-129:1, 154-155:1, 161-164:1, 167-168:1, 188-189:1)

on the cost-based approach [126]. The computation of the optimal operation problem in the original 365 days takes 6.3s, and the evaluation of the objective function obtains $66,003. In contrast, it takes 0.4s for the computation on the set of typical and extreme days, reaching the objective value $65,931. The computational efficiency is boosted by 16 times while the relative error stays about 0.1%.

4.5.2 Learning performance

Fig.4.6 reports the moving average curve over 500 episodes of the mean episode reward of the rainbow DRL agents during learning, benchmarked by the DDQN agent and Proximal Policy Optimization (PPO) agent. PPO is a state-of-the-art on-policy DRL algorithm with an actor-critic architecture. The shadowed areas denote the standard deviation of the performance over three random seeds. The candidate nodes are $\{2, 7, 26, 30\}$. Note that the negative value of the reward refers to the discounted total cost of the planning scheme scaled by 10^6. With the rise of the number of learning episodes, all four curves rise dramatically in the early phase and converge after about 8000 episodes, demonstrating the feasibility of the coordinated DRL-optimization framework. Thanks to the remarkable exploration capability, the rainbow curve transcends the DDQN and the PPO at about 1,500 and 500 episodes, despite a slightly worse initial point. The mean episode reward of the rainbow agent converges to 1.228, which is 9% and 6% better than those of the DDQN and PPO agents, respectively. Also, the rainbow agent exhibits impressive convergence speed (1.56×10^{-4} per episode) compared to its opponents (DDQN: 7.99×10^{-5}, PPO: 8.99×10^{-5}) during episode 0 to 5,000. This is attributed to the adopted N-step learning and PER technique, etc. The rainbow algorithm with quantile regression, marked as Rainbow-QR in Fig.4.6, achieves the best result among all algorithms. Compared with the original rainbow algorithm, the rainbow-QR has a better mean episode reward (-1.217) and smaller variance during learning, owing to removing the bound on the value distribution. For this planning problem, the Rainbow-QR agent takes approximately 11,000 seconds per 2000 episodes. Moreover, it can be seen from Fig.4.6 that after approximately 14,000 seconds (about the 2545th episode), the agent has already learned the optimal planning policies.

4.5.3 Verification of near-optimality and solution generation efficiency

This subsection carries out comparative experiments to demonstrate the effectiveness of the proposed DRL method in terms of both near-optimality and solution-generation efficiency. DRL, optimization solver, and Benders decomposition (BD) are applied to solve the MILP problem formulated in equations (4.1) to (4.5), assuming a static growth rate of future load/RES and fixed future battery prices (i.e., [345, 242, 198, 186, 174] in this case

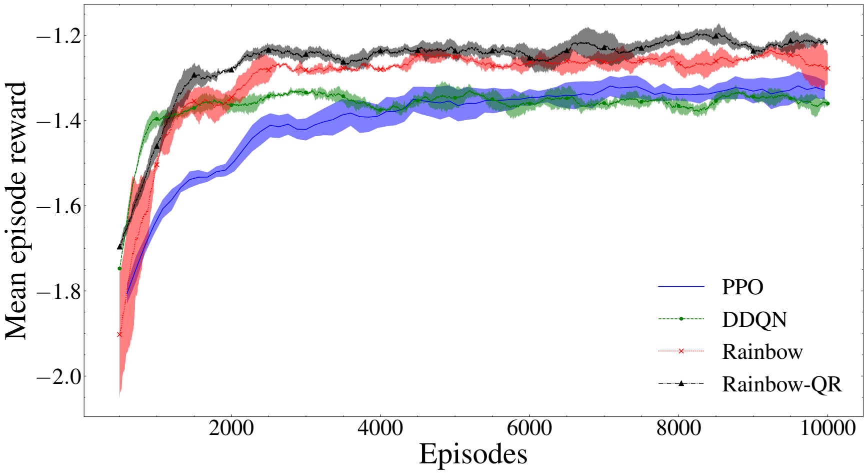


Figure 4.6: Various DRL learning performance intervals on episode reward

study). It is worth mentioning that, although these assumptions are used in this subsection, the DRL method is not limited to them, as the DRL method produces a stochastic planning policy rather than a fixed planning solution.

The implementation of the BD algorithm is based on [102]. In the BD approach, the master problem involves integer decision variables for battery capacity, location, and timing, and is an integer programming problem. The subproblem is an LP problem that only involves continuous decision variables for the microgrid's daily operation. Although the computational framework of BD and our proposed bi-level framework are similar in decoupling the decision-making of discrete and continuous variables, the proposed framework has the advantage that after offline learning, the upper and lower levels are decoupled during online inference, and only the upper layer that generates planning decisions is needed. On the other hand, the master problem and subproblem of the BD approach are closely intertwined, and these two problems cannot run independently.

The MILP solver approach based on the deterministic integrated formulation solves decisions for multiple time periods simultaneously. This MILP planning problem has 663,121 constraints and 572,896 variables, where 572,816 of them are continuous variables, and 80 of them are integer variables. When switching to the DRL method, an episode consisted of five steps. For the lower-level LP problem at a single step, there are approximately 74,760 constraints and 76,251 continuous variables. The average solving time for this is 0.6 seconds.

Table 4.3 compares the expansion decisions, total costs, and computational efficiencies of the DRL, MILP solver, and BD methods in three scenarios. The three scenarios are defined based on the magnitude of the net load growth rate. The table shows the planning decisions made for different years in a multi-period manner, starting from the 0th year and up to the 20th year. For example, in Scenario 1, DRL, solver, and BD recommend installing a 1000kWh BES at node 26 in the tenth year without the need to install BES at other time periods to save costs. It is observed that the expansion plans provided by these three methods are very similar, with completely consistent choices in terms of timing and capacity. They all tend to install at nodes 26 and 30, which are on the same branch of the test system. Additionally, the total costs consisting of investment and operational costs resulting from the planning solutions of these three methods are exactly the same. The higher net load in Scenario 3 also drives these three methods to install 500kWh of battery storage in the 15th year to hedge against potential power shortages. The efficiency of solution generation, ranked from highest to lowest, is DRL, MILP solver, and BD. Owing to the decoupling of the upper and lower levels, combined with the rapid matrix computation of neural networks, the DRL approach is over 40,000 times more efficient than the solver-based approach across all three scenarios. On the other hand, the computation time of BD is 3.7 to 5.6 times that of the MILP solver (i.e., Gurobi 9.5.1), and much higher than that of the DRL method. Although in the BD framework, the decomposition of the master problem and the subproblem reduces the computational complexity of each problem, the trade-off is the need for iterative computations that are time-consuming. Overall, these results confirm that the proposed DRL has a great potential for solving large-scale planning problems. It can significantly improve the efficiency of generating planning solutions while achieving optimization performance comparable to other model-based optimization techniques.

Discussion: A five-year interval in the case studies is chosen to highlight the impact of the decreasing cost of battery energy storage on the planning scheme. Based on the storage capital cost projection data obtained from [3], the reduction in cost every five years is significant, which can have a huge impact on planning decision and help showcase the effectiveness of the proposed approach. It is noted that the proposed planning framework can

Table 4.3: Comparison of Expansion Decisions, Costs, and Computation Time for DRL and MILP.

		Scenario I (low net load)			Scenario II (intermediate net load)			Scenario III (high net load)		
RES annual growth rate		5%			2%			1%		
Load annual growth rate		3%			2%			2.5%		
Method		DRL	MILP solver	BD*	DRL	MILP solver	BD*	DRL	MILP solver	BD*
Multi-period expansion decisions Format: capacity (kWh) @ node, - indicates no installation	0th year	-	-	-	-	-	-	-	-	-
	5th year	-	-	-	-	-	-	-	-	-
	10th year	1000@26	1000@26	1000@26	1000@26	1000@26	1000@30	1000@26	1000@30	1000@30
	15th year	-	-	-	-	-	-	500@26	500@26	500@30
	20th year	-	-	-	-	-	-	-	-	-
Total Cost (10^6\$)		1.69	1.69	1.69	1.49	1.49	1.49	1.81	1.81	1.81
Time required for generating decisions (s)†		0.004	167.032	total: 776.772 master: 14.776 sub: 761.996	0.004	226.467	total: 834.719 master:14.913 sub: 819.806	0.004	293.337	total: 1635.480 master: 48.831 sub: 1586.648

* BD: benders decomposition; †Average over three independent runs.

fully accommodate other decision intervals, e.g., 1 or 3 years, with almost no modifications needed.

4.5.4 Scalability

The next part of the experiment attempts to investigate the scalability and agility of the proposed method and MILP method by gradually increasing the number of candidate nodes, which amounts to increasing the dimension of feasible policy space. Fig.4.7 plots the training curves for planning problems with four ($\{2, 7, 26, 30\}$), six ($\{2, 7, 21, 24, 26, 30\}$), eight ($\{2, 7, 12, 15, 21, 24, 26, 30\}$), and ten($\{2, 4, 6, 7, 12, 15, 21, 24, 26, 30\}$) candidate nodes, respectively. Comparing the results in Fig.4.7, it can be seen that these four curves performed differently before 4000 episodes, but after that, they almost converged to the same planning policy. The results before 4000 episodes reveal that the agent with a smaller feasible policy space has a better initial solution and can converge to the optimal policy faster in general. The comparisons are also conducted in the deployment phase of the trained DRL agent to evaluate the scalability of the computation efficiency, which are shown in Table 4.4. The mean and standard deviation of the results are derived from three independent runs. As the number of candidate nodes increases from four to ten, the time consumption for the DRL agent to formulate decisions escalates by 61%. In contrast, its MILP counterpart witnesses

Table 4.4: Computation efficiency of the DRL and MILP approach.

Method	Number of candidate nodes			
	4	6	8	10
DRL	4.16 ± 0.11 ms	4.37 ± 0.10 ms	5.03 ± 0.28 ms	6.68 ± 0.18 ms
MILP solver	118.7±5.0 s	229.7±1.5 s	438.3±2.5 s	673.3±11.0 s

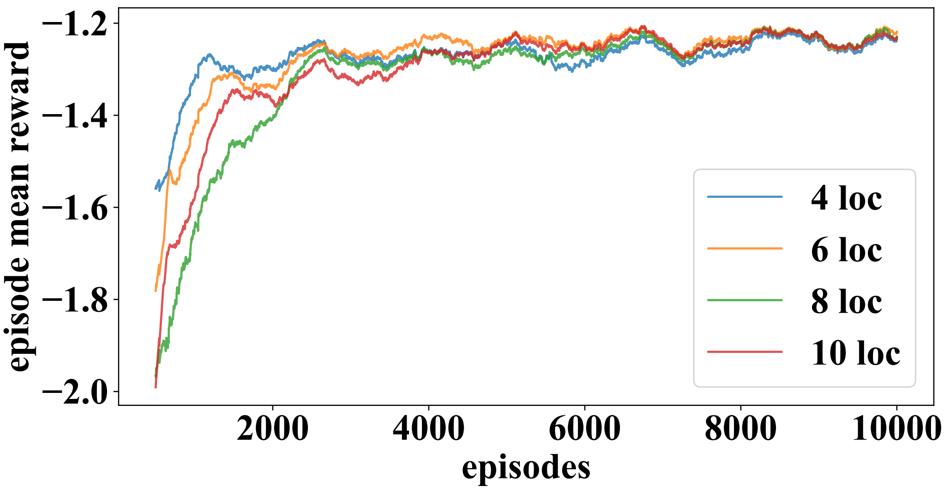


Figure 4.7: Training curves of different numbers of candidate nodes.

a surge of 466%. More candidate locations result in a larger action space of the DRL agent, which then leads to an increase in the size of the matrix computation involved in the NNs. Table 4.4 and Fig.4.7 jointly demonstrate the proposed DRL BES expansion framework is agile and scalable.

4.5.5 Generalizability to uncertain future scenarios

The use of DRL provides a more flexible and robust solution to storage planning compared to traditional optimization techniques. Most optimization techniques provide a deterministic and fixed planning solution and assume that this planning solution can adapt to all future scenarios. In contrast, DRL can continuously learn from experience and adapt to changing circumstances by providing a dynamic and stochastic planning policy. This section demonstrates that the DRL agent can adapt to new and unforeseen scenarios without requiring additional learning.

For illustration purposes, Table 4.5 enumerates five future trajectory sets alongside their respective BES expansion strategies. Scenario #0 serves as a control group herein. The decision-making process is carried out in a rolling manner. The rolling decision process adds flexibility to the decision-making because the planning scheme is not determined all at once, but dynamically adjusted over time as uncertain factors change. For example, consider scenarios 0 and 1 in Table IV. In Scenario #0, the future energy storage prices are [345, 242, 198, 186, 174], while in Scenario #1, they decrease significantly to [345, 212, 143, 129, 115]. In Scenario #1, the energy storage prices decrease by 0%, 12%, 28%, 31%, and 34%, respectively, compared to Scenario #0, for each decision interval. In Scenario #1, upon encountering a price of 129 in the fourth time step, the DRL agent discerns the price drop's magnitude surpassing previous instances. As a result, it opts to introduce an added 500kWh of energy storage immediately to capitalize on this price reduction. The DRL agent also has a long-term perspective, knowing that even though the price drop is the largest in the fifth time step, installing storage in the fourth time step can allow it to enjoy the benefits of energy storage as early as possible, reducing the need to curtail RES and load shedding during the fourth time slot. In contrast, the opposite suggestion is given for Scenario #2, which is a pessimistic scenario. The installation in the fourth period is canceled due to the high investment cost. Since the prices stay unchanged, the agent judiciously suggests the installation in the third period rather than the fourth period to take advantage of the BES units earlier.

The impacts of different generation adequacy levels are investigated in Scenario #3 and Scenario #4, which represent the scenarios with and without adequate power supply, respectively. In Scenario #3, no BES unit is proposed to be built. On the contrary, an additional 500kWh BES is proposed to install in the fourth period relative to the baseline. Moreover, due to the massive development of RES, the MG chooses to export power to the external grids at all times in Scenario #3, which earns $349,373. However, for Scenario #4, the MG has to spend $634,402 on importing power.

Table 4.5: Planning solutions provided by the DRL agent under various scenarios.

Scenario	L^{price}	L^{res}	L^{load}
#0: Baseline	[345, 242, 198, 186, 174]	[1, 1.1, 1.2, 1.25, 1.42]	[1, 1.2, 1.3, 1.4, 1.5]
#1: Storage price decrease	[345, 212, 143, 129, 115]	[1, 1.1, 1.2, 1.25, 1.42]	[1, 1.2, 1.3, 1.4, 1.5]
#2: Storage price increase	[345, 345, 345, 345, 345]	[1, 1.1, 1.2, 1.25, 1.42]	[1, 1.2, 1.3, 1.4, 1.5]
#3: RES increase, load decrease	[345, 242, 198, 186, 174]	[1, 1.2, 1.4, 1.6, 1.8]	[1, 1.05, 1.1, 1.15, 1.2]
#4: RES decrease, load increase	[345, 242, 198, 186, 174]	[1, 1.05, 1.1, 1.15, 1.2]	[1, 1.2, 1.4, 1.6, 1.8]

Scenario	Scheme {location: capacity (kWh)}					Total Cost (10^6\$)
	0th year	5th year	10th year	15th year	20th year	
#0: Baseline	-	26@1000	26@500	30@500	-	1.75
#1: Storage price decrease	-	26@1000	30@500	26@1000	-	1.71
#2: Storage price increase	-	30@1000	26@500	-	-	1.86
#3: RES increase, load decrease	-	-	-	-	-	1.03
#4: RES decrease, load increase	-	26@1000	26@500	26@1000	-	2.38

4.5.6 Planning scheme analysis

This subsection analyzes the planning schemes produced by the proposed approach. Without loss of generality, future scenarios are set to $L^{\text{res}} = [1, 1.1, 1.2, 1.25, 1.42]$, $L^{\text{ld}} = [1, 1.2, 1.3, 1.4, 1.5]$, and $L^{\text{price}} = [345, 242, 198, 186, 174]$, for demonstration. The candidate nodes are $\{2, 7, 21, 24, 26, 30\}$.

Firstly, TABLE 4.6 shows the solutions generated by the agent at the upper level after different learning episodes. The trend suggests that as the agents undergo learning, the installation locations transition from heterogeneous to homogeneous, with the total installed capacity decreasing and the installation time becoming more dispersed. The three solutions generated in TABLE 4.6 are used to evaluate the impact of the upper-level decisions on the MG operation at the lower-level. The cost of the operation sub-problem, as shown in (4.6), is designated as the operation indicators, which consist of costs such as purchased power, primary reserve, storage operation, load shedding, and DG operation. Primary reserve cost and load shedding cost are indicators closely related to the safety of MG operation. The former refers to the cost of the backup capacity of DGs prepared for islanding events, while the latter refers to the penalty cost incurred due to insufficient power supply adequacy. Fig.4.8 exhibits the impact of upper-level decisions on the operational indicators of the

Table 4.6: Solutions generated by the agent at the upper level after different learning episodes.

DRL Agent	Multi-period expansion decisions Format: capacity (kWh) @ node - indicates no installation				
	0th year	5th year	10th year	15th year	20th year
Random	2000@7	1000@24	1000@24	2000@21	1000@26
Agent_1000	-	500@30	-	2000@21	-
Agent_4000	-	1000@26	500@26	500@30	-

Note: Random, Agent_1000, Agent_4000 refer to DRL agents that have not undergone learning, undergone 1000 episodes of learning, and undergone 4000 episodes of learning, respectively.

lower-level. Note that the costs presented in Fig.4.8 are discounted by the discount rate. The random agent suggests installing 7000kWh BES in total. Such large battery capacity, along with its flexibility, results in the lowest operating cost of $1,403,003. This scheme also performs the best in terms of safety indicators, i.e., the sum of primary reserve cost and load shedding cost, which is the lowest at $4,035. However, when considering the investment cost, the high investment cost of the random agent ($1,307,970) makes it the most uneconomical and impractical solution. The investment cost of the solution proposed by Agent_1000 is the lowest, as it chooses to install 2000kWh of BES in the 15th year when the battery price is low. However, due to its poor location selection, its operating cost and safety indicators are the worst, requiring a significant amount of safety-related cost ($354,281). Agent_4000 represents a well-trained agent. Starting from the second period, this agent suggests that 1000kWh, 500kWh, and 500kWh BES units be installed on nodes 26, 26, and 30, respectively, three periods in a row. It can strike a balance between investment cost and operating cost, achieving a 29% reduction in safety-related cost and a 15% reduction in total cost with only an 8% higher investment cost compared to Agent_1000.

Table 4.7 examines the detailed expenditure of the planning scheme by Agent_4000. It is observed from Table 4.7 that almost all costs rise with the increase of load except for the cost of load shedding. This expansion scheme substantially reduces the cost of load shedding in the third and fourth periods, despite the fact that the amount of load is increasing.

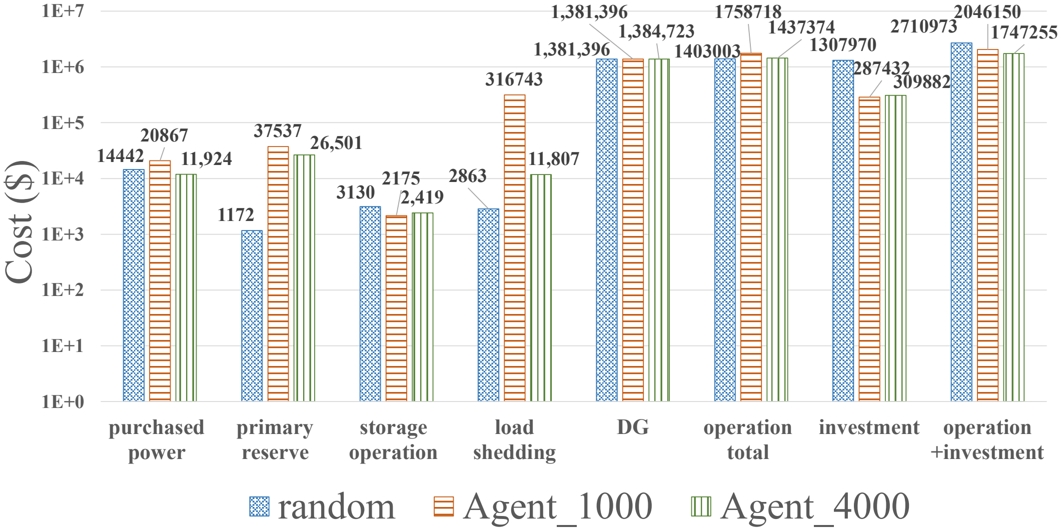


Figure 4.8: The impact of upper-level decisions on the operational indicators of the lower-level.

Table 4.7: BES Expansion Scheme and Associated Expenditure.

Scheme	Investment Cost ($)	Operation Cost ($)					Total Cost ($)
		DG	External	Reserve	Storage	Load Shedding	
0th year: -	-	345,216	-85,459	4,257	0	0	264,013
5th year: 1000@26	242,000	408,824	-16,290	5,789	256	1,582	400,162
10th year: 500@26	99,000	430,704	30,039	7,338	552	0	468,634
15th year: 500@30	93,000	447,834	82,893	9,819	1,526	226	542,298
20th year: -	-	460,114	136,022	19,148	2,732	26,259	644,276

Note: negative value in External column denotes the revenue of selling power to external grids via the PCC.

The benefit and necessity of the expansion of BES in the MG are illustrated in Table 4.8, which shows that the BES units reduce load shedding and facilitate the integration of RES simultaneously. Also, the MG tends to purchase power from the external grids as the load increases. This leads to a rise in the exchange power at the PCC, which in turn requires the DG units to prepare more reserve capacity as enforced in (4.17) and (4.22). The timely expansion of the BES mitigates the need for reserve capacity, resulting in the improvement of cost-effectiveness.

Table 4.8: Comparison of ratios of load shedding and RES curtailment between solutions with and without storage installed.

Solution	Type	Ratio (%)				
		Year 0-4	Year 5-9	Year 10-14	Year 15-19	Year 20-24
With storage	RES curtailment	0.000	0.000	0.000	0.182	0.377
	Load shedding	0.000	0.003	0.000	0.001	0.058
Without storage	RES curtailment	0	0.012	0.051	0.126	0.775
	Load shedding	0.000	0.320	1.250	2.180	3.280

The DRL planning framework is employed to solve the expansion problems with and without frequency dynamics constraints (4.16) to (4.23), respectively, thereby appraising the impacts of these constraints. We refer to the former as the islanding-safe problem and the latter as the islanding-unsafe problem. As for the islanding-unsafe problem, the DRL approach suggests installing 500kWh on node 26 in the second, third, and fifth periods, respectively, which is 500kWh less than that of the islanding-safe problem. The incorporation of frequency-induced constraints drives the construction of more BES units so that the MG can remain resilient during islanding events. In the meanwhile, the consideration of these safety-related constraints leads to an increase in the cost. The DRL approach attains the discounted total cost 1.75×10^6 in the planning problem with frequency-induced constraints and 1.64×10^6 without them. This finding is consistent with the expected value of the discounted total cost over various RES/load/price trajectories while training the DRL, i.e., the episode mean reward. The expected value is 1.17×10^6 for islanding-unsafe problems and 1.23×10^6 for islanding-safe problems, indicating a 5% cost rise.

4.6 Summary

A DRL-aided multi-period BES expansion strategy for MG considering the islanding contingency is proposed in this study, which features a bi-level framework. A series of comparative experiments has been designed and conducted on a 33-bus MG to examine the performance of the proposed planning strategy. Compared to the DDQN and PPO peers, the adopted rainbow algorithm with quantile regression exhibits better convergence speed and mean episode reward. The DRL-based planning approach can deliver near-optimal solutions with more than 40,000x speed improvement relative to the solver-based approach. The increase in the number of candidate nodes has a minor impact on computational time and little effect on the solution quality, which substantiates the scalability of the proposed framework. The experiments also verify that the DRL approach can agilely generalize to uncertain future scenarios without extra effort. For future work, one can leverage the idea of transfer learning to reduce the demand for training data [127, 128].

Chapter 5
Quantum Policy Learning for Energy Storage Arbitrage

This section explores the potential of quantum computing to elevate energy arbitrage task efficiency in energy storage systems. The research introduces a Quantum Policy Learning algorithm that harnesses a variational quantum circuit alongside a devised Markov decision process, aiming to optimize online energy arbitrage processes in the presence of quantum hardware noise. Empirical analyses highlight the quantum advantage of the method in terms of rapid convergence and optimization, demonstrating its comparative edge over conventional methods including classical reinforcement learning and model predictive control.

5.1 Introduction

In the evolving landscape of electric networks, energy storage systems (ESSs) emerge as essential components, playing a transformative role in enhancing power system flexibility [129]. ESSs demonstrate their significance through a wide spectrum of services, from peak-shaving to frequency regulation and accommodating renewables. Among the various services ESSs provide, energy arbitrage (EA) is often considered one of the more essential and potentially profitable for ESS owners. EA leverages the daily fluctuations in energy prices, purchasing energy during low-demand periods and capitalizing on its value during peak times. The EA for ESSs bolsters the efficiency of grid operation and impacts the electricity market dynamics, offering a symbiotic advantage for both ESS owners and grid operations.

Within the EA for ESSs domain, two primary avenues of research have attracted attention: optimization-based techniques and the burgeoning application of reinforcement learning (RL) techniques. Optimization-based techniques have long been the cornerstone of the EA for ESSs research. When assuming complete certainty in electricity price information, as is often the case in certain planning scenarios where ESS profitability is validated against historical data, EA is typically modeled as either a mixed-integer linear programming (MILP) or mixed-integer nonlinear problem. These models are subsequently tackled using dedicated solvers or heuristic algorithms [130,131], respectively. As the operation in a shorter time scale contends with price volatility, the focus shifted towards stochastic models [132]. Leveraging a dynamic programming approach, the month-long EA for ESSs problem is decomposed into daily, hourly, and real-time segments, ensuring both reduced complexity and solution optimality in [133]. Ref. [134] emphasizes the co-optimization of energy storage with objectives aimed at EA and local power factor correction. It introduces a model predictive control (MPC)-based scheduling policy, incorporating auto-regressive forecasts to handle uncertainties adeptly. The fluctuation inherent in electricity prices led to the exploration of robust optimization. When faced with increasing forecast errors, robust optimization strategies demonstrated superior economic performance compared to deterministic methods [135].

Optimization-based methods, while robust, have inherent limitations. For electricity price prediction errors characterized by multiple scenarios or distribution information, optimization-based methods often require multiple modeling or solving attempts, resulting in a specific solution. This not only creates computational overhead but also demands a delicate balance between the granularity of the model and its solvability. In contrast, RL techniques can bypass these complexities, offering the ability to swiftly adapt to a multitude of unforeseen scenarios after just one offline training session. The model-free feature is another reason why RL techniques is increasingly valued. The study presented in [41] utilizes a model-free deep reinforcement learning (DRL) method to optimize EA for ESSs, incorporating a precise battery degradation model. The empirical testing on U.K. wholesale market data showcases its superiority over model-based MILP approaches. Ref. [136] leverages the Rainbow Deep Q-Networks to oversee battery operations within a microgrid, aiming to achieve efficient EA and enhance the utilization of solar and wind power. Ref. [34] introduces a Proximal Policy Optimization agent tailored for continuous action spaces to execute the capacity scheduling of a hybrid ESS. Validated within the PJM market, this approach underscores that employing a continuous action space via DRL can refine the scheduling outcomes, leading to enhanced profitability. Both optimization and RL methodologies have advanced in the EA domain, yet there are still existing gaps. Traditional optimization techniques often struggle with model complexity and scalability issues, especially in complex and dynamic energy markets. Classical RL paradigms face challenges in efficiently handling vast datasets [137]. Additionally, these paradigms face complexities in training tasks in complex and real vector spaces, which involves optimizing numerous trainable parameters inherent in the neural network representation.

Quantum computing (QC), grounded in the principles of quantum mechanics, is a disruptive computational paradigm. Contrary to classical bits, which can only be in either a 0 or 1 state, quantum bits (qubits) can simultaneously occupy both states due to superposition, giving QC a unique advantage. This advantage is further enhanced by quantum entanglement, where multi-qubit systems show correlations that exceed those found in classical systems. Representative algorithms in QC, such as Grover and Shor algorithms, demonstrate quantum advantages that are unmatched by their classical counterparts. The integration of QC with machine learning birthed quantum machine learning (QML). Pioneering algorithms in QML, spanning from quantum discriminative [138] to generative networks [139], are characterized by the application of differentiable quantum circuits, i.e., variational quantum circuits (VQCs). In the study presented by [140], the capabilities of VQCs are meticulously explored to tackle the transient stability assessment in electrical power systems. Experimentation on quantum simulators and real-world quantum computing platforms substantiate its superior accuracy, resilience to noise, and scalability compared to traditional machine learning methodologies.

As the potential of VQCs has been gradually validated both theoretically and empirically [140,141], efforts have begun to integrate VQCs into the RL paradigm [142–144]. While DRL has seen substantial advancements in diverse applications, quantum reinforcement learning (QRL) remains relatively nascent, particularly in energy systems applications. Ref. [142] is the first proof-of-principle demonstration of VQCs to approximate the Q-value function for decision-making. The QRL introduced in Ref. [142] has predominantly revolved around discrete control, which is subsequently tested in a toy text environment named frozen-lake. Transitioning from discrete to continuous actions, the spectrum of QRL applications can be expanded. Recent advancements include the variational quantum soft actor-critic (SAC) [142], which proposes hybrid quantum-classical solutions for the cartpole problem, which is a continuous control task. A quantum policy gradient algorithm is presented in [144]. However, the results are still limited to toy environments, including contextual bandits, cartpole, and frozen-lake.

The potential of QRL in transforming the landscape of decision-making processes is compelling. However, several significant challenges and gaps in its current research landscape need to be addressed. Even though some efforts have been dedicated to exploring classical control tasks, their applications to practical engineering problems, especially in the energy sector, remain largely unexplored. Ref. [145] emerges as a pioneer, which utilizes the QRL in the distributed frequency control of islanded microgrids. Secondly, the natural discreteness inherent to quantum systems hinders the development of QRL for continuous action spaces without careful crafting and design. More importantly, the current body of research lacks comprehensive testing on actual quantum hardware, resulting in a limited understanding of potential errors and a lack of insight into real-world robustness. For example, the study in [145] relies solely on quantum simulators for testing and validation. Consequently, the practicality of QRL applications in energy systems, along with their performance on noisy hardware during the noisy intermediate-scale quantum era (NISQ), necessitates more in-depth scrutiny [146].

Driven by the opportunities and challenges aforementioned, this work aims to harness the power of QRL and VQC to address the EA for ESSs. A Quantum Policy Learning (QPL) algorithm, which is entirely based on a quantum circuit, is devised to perform the decision-making in the EA in a sequential and online way. State vectors, including real-world ESS status and price signals, are embedded into the Hilbert space through angular embedding and data-reuploading scheme. The output derived from the QPL agent provides the expected value and variance of optimal ESS charging/discharging capacities. To safeguard the robustness of the QPL against the potential noise intrinsic to quantum hardware, a VQC with low depth and low width, yet possessing adequate expressive power, is designed. The proposed VQC, trained using a hybrid quantum and classical manner, is subsequently deployed on IBM's quantum computing platform. Through this deployment, a comprehensive evaluation is conducted, assessing not only the decision-making capabilities of the QRL agent but also its quantum advantage in higher expressivity, resilience to noise, and practical applicability.

The contributions of this paper are summarized as follows:

1. Inspired by and surpassing the capabilities of classical DRL methods prevalent in ESS EA approaches, this article proposes a QPL approach to alleviate the complexity of the decision-making policy and secure enhanced profits. The introduced QPL utilizes a hybrid classical-quantum gradient update paradigm during the training phase. Once trained, the quantum policy agent can be deployed on real quantum hardware to efficiently undertake online EA tasks.

2. The proposed QPL collaborates with a strategically devised Markov Decision Process (MDP) to facilitate the accumulation of experiences through the promotion of early-stage proactive exploration of the action space. In navigating the challenges presented by the noisy quantum hardware, our work devises the VQC within the QPL framework with distinctive low depth and width characteristics. The distinctive VQC structure incorporates alternating variational entanglement sub-layers and variational embedding sub-layers to span over the Hilbert space adequately. In the process of optimizing the quantum policy parameters, this study derives a well-formulated loss function and gradient computations, which can integrate advanced action post-processing techniques, including re-parameterization, squashing, and linear transformations.

3. In comparison to classical DRL approaches, optimization techniques, and Model Predictive Control (MPC) methodologies, as well as other VQC architectures, the proposed QPL achieves superior optimization performance and accelerated convergence rates, all while maintaining a lower model complexity. The following tests on IBM_Lagos quantum hardware validate the noise resilience of the proposed QPL method.

The remainder of this paper is organized as follows. Section 5.2 presents a Markov decision process for energy storage arbitrage. A classic entropy-regularized DRL algorithm named Soft Actor-Critic is introduced in Section 5.3. Section 5.4 discusses the fundamentals of quantum computing and variational quantum circuits sequentially; followed by proposing a quantum variational circuit with low depth and width as the policy network for QPL, and derives the loss function for training this specific VQC. The numerical results are exhibited in Section 4.5, followed by the concluding remarks in Section 5.6.

5.2 Markov decision process for Energy Storage Arbitrage

This section formulates a specific MDP for the EA for ESSs. The inherent Markovian properties of MDPs align well with the Markovian characteristics of energy storage operations, especially when considering the stochastic nature of energy prices and storage dynamics. The design principle focuses on optimizing financial returns while accounting for the operational constraints of ESSs and uncertainties intrinsic to price signals.

The MDP can be characterized by a 5-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}(s_{t+1} = s' \mid s_t = s, a_t = a), r(s,a), \gamma)$, wherein $\mathcal{S}$ represents the state space, $\mathcal{A}$ denotes the action space, $\mathcal{P}$ signifies the transition probability between states, $r(s,a)$ is the associated reward function, and γ stands for the discount factor applied to future rewards. The design for each component is further elucidated in the subsequent subsections.

5.2.1 State Space

States represent the pertinent resources and information required for making decisions. States are expected to have a strong correlation with decisions and rewards, enabling the agent to infer the optimal strategy based on the state. In this task, the state vector is defined as:

$$\mathbf{s_t} := (SOC_t, t, P_{\mathrm{lmp},t}, \mathcal{C}_t), \tag{5.1}$$

where SOC_t denotes the State of Charge of the ESS at time t. It provides information about the available energy in the system, which is crucial for decision-making regarding energy dispatch or charging. t represents the current time period. Including the time as a state variable can be useful, especially in problems where policies or decisions might vary based on the time of day. $P_{\mathrm{lmp},t}$ is the locational marginal price of electricity at time t. This price signal is essential for determining optimal energy trading strategies, as it indicates the cost or revenue associated with buying or selling energy at a particular time.

$\mathcal{C}_t$ represents the cost associated with the energy currently stored in the ESS at time t. One distinctive feature of the MDP model presented in this paper is the explicit incorporation of $\mathcal{C}_t$. By incorporating this variable, historical procurement costs of energy are accounted for, ensuring that the selling price consistently covers the initial purchase costs and any associated overheads. This allows for a more realistic and economically viable arbitrage strategy, particularly in volatile energy markets where prices can fluctuate significantly. The updating and maintenance of $\mathcal{C}_t$ will be elaborated upon in the subsequent section on state transitions.

5.2.2 Action Space

For a single ESS, the agent operates within a continuous action space, facilitating more refined dispatch decisions and augmenting profit-making potential compared to a discrete action space. The dimension of this action space is one-dimensional, representing the charging or discharging power. While the standard configuration operates on an hourly basis, the action interval is not confined to one hour and can be tailored according to the market mechanism governing the ESS operations. The sign of the action indicates the operation mode: positive for charging and negative for discharging, with the magnitude representing the power level. Typically, the action of RL is mapped to a [-1,1] interval through the *tanh* function. Multiplying this action by the maximum power capacity produces a ready-to-use battery dispatch signal, as expressed in the following formula:

$$p_t' = a_t \times P_{\max}, \tag{5.2}$$

where p_t' signifies the preset battery dispatch power, a_t represents the action output from the RL agent within the range [-1,1], and $P_{\max}$ is the maximum power capacity of the ESS.

5.2.3 State Transition

The transitions between states in the MDP capture the dynamics and randomness of the system. This subsection elaborates on the patterns of state transitions across time steps.

In Eq. (5.1), the transition of $P_{\text{lmp},t}$ depends on the clearing mechanism of the electricity market and the supply-demand relationship. Assuming the ESS plays the role of a price taker in the market, $P_{\text{lmp},t}$ is not influenced by the decisions of the ESS, and thus, the ESS can directly obtain the data from the market. The transition of t is more straightforward: it increases by one with each time step.

The transitions of SOC_t and $\mathcal{C}_t$ are closely related to the action, that is, when and how much capacity to charge or discharge.

$p'_t < 0$ signifies discharging of the ESS. Examining if the SOC decline due to the preset discharge power will cause the SOC to fall below the threshold:

$$\begin{cases} \textbf{if } SOC_t + \frac{p'_t \cdot \delta_h}{\eta_{\text{dis}} \cdot U} \geq \underline{SOC} : \\ SOC_{t+1} = SOC_t + \frac{p_t \cdot \delta_h}{\eta_{\text{dis}} \cdot U}, p_t = p'_t, \kappa_t = 0 \\ \textbf{otherwise} : \\ SOC_{t+1} = SOC_t, p_t = 0, \kappa_t = \kappa_0, \end{cases} \tag{5.3}$$

where p_t is the actual discharging/charging power at time t, δ_h represents the time interval, indicating the duration over which the discharge action is applied (1 hour in case studies), η_{dis} is the discharging efficiency of the ESS, U denotes the energy capacity of the ESS, and $\underline{SOC}$ is the minimum allowable SOC for the ESS. κ is the penalty applied when certain safety or operational constraints are violated in the system. The introduction of κ can ensure that the RL agent adheres to the desired operational boundaries and avoids actions that may harm the system. κ_0 denotes a predefined penalty magnitude, serving as a guideline for the severity of the constraint violation.

Through capturing the energy discharged during the current time step, the cumulative cost associated with the energy stored in the ESS up to time $t + 1$ is decreased by:

$$\mathcal{C}_{t+1} = \mathcal{C}_t + \mathcal{C}_t \cdot \frac{p_t \cdot \delta_h}{\eta_{\text{dis}} \cdot U \cdot SOC_t}. \tag{5.4}$$

$p'_t > 0$ signifies charging of the ESS. Inspecting whether the increase in SOC, due to the predetermined charge power, will lead the SOC to climb above the set threshold:

$$\begin{cases} \textbf{if} \;\; SOC_t + \frac{p_t \cdot \delta_h \cdot \eta_{\text{ch}}}{\cdot U} \leq \overline{SOC} : \\ SOC_{t+1} = SOC_t + \frac{p_t \cdot \delta_h \cdot \eta_{\text{ch}}}{\cdot U}, p_t = p'_t, \kappa_t = 0 \\ \textbf{otherwise:} \\ SOC_{t+1} = SOC_t, p_t = 0, \kappa_t = \kappa_0, \end{cases} \tag{5.5}$$

where $\overline{SOC}$ is the maximum allowable SOC for the ESS.

The incremental cumulative cost due to charging behavior can be derived by:

$$\mathcal{C}_{t+1} = \mathcal{C}_t + p_t \cdot \delta_h \cdot P_{\text{lmp},t}. \tag{5.6}$$

5.2.4 Reward Function and Reward Discount Factor

Unlike most EA modeling in MDP, where the cost or revenue from electricity trading in the current time step is directly considered as the reward, this study devises a reward function that encourages the RL agent to thoroughly explore the charging and discharging action space.

It is believed that profits/losses are settled only when the ESS discharges. The act of charging is to create room for future arbitrage. Therefore, charging itself is merely a transformation and not the actual cause of loss. With the discharge process, the ESS accrues revenue, which is calculated as $-p_t \cdot \delta_h \cdot P_{\text{lmp},t}$. Considering the cumulative cost associated with the discharging capacity, the net gain of the ESS when discharging at time step t is:

$$g_t = -p_t \cdot \delta_h \cdot P_{\text{lmp},t} + \mathcal{C}_t \cdot \frac{p_t \cdot \delta_h}{\eta_{\text{dis}} \cdot U \cdot SOC_t}, \tag{5.7}$$

where the second term denotes the cost associated with the energy transacted during the current time step.

Summing up all gain and penalty, the reward function can be described as

$$r_t = g_t - \kappa_t - deg_t, \tag{5.8}$$

where deg_t represents degradation cost, which is proportional to the absolute value of dispatched power, as given by $deg_t = |p_t| \cdot \mathrm{c}$; c is a constant degradation cost per unit of power.

Quantifying the charging behavior as a negative reward may lead the agent to prematurely converge to a local minimum where it neither charges nor discharges. This is because, during the early exploration phase, the agent often outputs charging actions, resulting in a majority of samples with negative rewards. Taking no action becomes an easily attainable and tempting local optimal strategy. Eq. (5.8) mitigates this issue to some extent.

The reward discount factor enables DRL agents to adjust their behavior, focusing on either long-term or short-term goals. The reward discount factor is set as a scalar γ smaller than one in this study.

5.3 Soft Actor-Critic

DRL algorithm is a state-of-the-art method commonly used to solve well-designed MDPs. As a DRL method, SAC originated from the work presented in Haarnoja et al.'s 2018 publication [147]. The core principle of SAC is entropy regularization. SAC trains the policy to strike a balance between anticipated returns and entropy, which quantifies the unpredictability of the policy. This section aims to briefly introduce the essence of SAC, leading to the forthcoming introduction of the QPL algorithm, which is derived from a portion of SAC.

As a member of entropy-regularized RL, SAC is distinguished from other RL methodologies in that the SAC agent receives an additional reward based on the entropy of its policy at every timestep. Hence, SAC attempts to train the following policy:

$$\pi^* = \arg\max_{\pi} \underset{\tau\sim\pi}{\mathrm{E}}\left[\sum_{t=0}^{\infty} \gamma^t \times \left(r\left(s_t, a_t, s_{t+1}\right) + \vartheta H\left(\pi\left(\cdot \mid s_t\right)\right)\right)\right] \tag{5.9}$$

where π^* represents the optimal policy that SAC aims to learn; τ is a trajectory sampled under policy π; ϑ is the temperature parameter, which modulates the trade-off between maximizing expected reward and increasing policy entropy; E denotes the expectation of the distribution; $H(\cdot)$ is the entropy term and is defined by $H(P) = \underset{x\sim P}{\mathrm{E}}[-\log P(x)]$, where P is the probability density function.

Under the SAC framework, the action value function $Q^\pi(s,a)$ is redefined as follows:

$$Q^\pi(s,a) = \underset{\tau\sim\pi}{\mathrm{E}}\left[\sum_{t=0}^{\infty} \gamma^t r\left(s_t, a_t, s_{t+1}\right) + \alpha \sum_{t=1}^{\infty} \gamma^t H\left(\pi\left(\cdot \mid s_t\right)\right) \mid s_0 = s, a_0 = a\right] \tag{5.10}$$

The Bellman equation captures a recursive relationship for the value of a state/action pair (s, a), based on the expected future value from its subsequent state/action pair (s', a'):

$$Q^{\pi}(s, a) = \underset{\substack{s' \sim P \\ a' \sim \pi}}{\mathrm{E}} \left[r\left(s, a, s'\right) + \gamma\left(Q^{\pi}\left(s', a'\right) - \alpha \log \pi\left(a' \mid s'\right)\right)\right] \tag{5.11}$$

where the right-hand side represents an expectation, with a subsequent state sourced from the replay buffer and a subsequent action derived from the existing policy. The Monte Carlo approximation of the above formula can be expressed as:

$$Q^{\pi}(s, a) \approx r + \gamma\left(Q^{\pi}\left(s', \tilde{a}'\right) - \alpha \log \pi\left(\tilde{a}' \mid s'\right)\right) \tag{5.12}$$

where $\tilde{a}' \sim \pi\left(\cdot \mid s'\right)$.

In SAC, there are three networks to be trained: one is π_θ which is parameterized by θ and represents the policy network; the remaining two are the action value functions Q_{ϕ_i} parameterized by ϕ_i, where $i = 1, 2$. Additionally, there are two target action value functions $Q_{\phi_{\mathrm{targ}, j}}$, where $j = 1, 2$. They are the delayed versions of Q_{ϕ_i}. The use of the double-Q trick and target Q can mitigate the overestimation of the Q-function and provide training stability in contexts where deep neural networks are used as function approximators and in environments with noisy rewards by reducing correlations rooted in the Bellman equation.

Temporal difference (TD) learning is employed to train the Q-networks. A mean squared TD error is used in the loss functions for Q-networks L:

$$L\left(\phi_i, \mathcal{D}\right) = \underset{\left(s, a, r, s', d\right) \sim \mathcal{D}}{\mathrm{E}} \left[\left(Q_{\phi_i}(s, a) - y\left(r, s', d\right)\right)^2\right] \tag{5.13}$$

where d indicates whether s' is a terminal state; $\mathcal{D}$ is a replay buffer; $y(r, s', d)$ is the TD target value for the Q-networks.

The target is given by deriving actions from the policy for the next state rather than using the actions from the samples:

$$y\left(r, s', d\right) = r + \gamma(1 - d)\left(\min_{j=1,2} Q_{\phi_{\mathrm{targ}, j}}\left(s', \tilde{a}'\right) - \vartheta \log \pi_\theta\left(\tilde{a}' \mid s'\right)\right) \tag{5.14}$$

In every state, the policy is supposed to optimize both the anticipated future rewards and the expected future entropy. By incorporating the minimum of the two Q networks and reparameterization trick, the policy loss can be written as:

$$\max_{\theta} \underset{\substack{s\sim D \\ \xi\sim\mathcal{N}}}{\mathrm{E}}\left[\min_{j=1,2} Q_{\phi_j}\left(s, \tilde{a}_\theta(s,\xi)\right) - \vartheta \log \pi_\theta\left(\tilde{a}_\theta(s,\xi) \mid s\right)\right], \tag{5.15}$$

where action $\tilde{a}_\theta$ is derived from a squashed Gaussian policy with an independent standard Gaussian noise ξ:

$$\tilde{a}_\theta(s,\xi) = \tanh\left(\mu_\theta(s) + \sigma_\theta(s) \odot \xi\right), \quad \xi \sim \mathcal{N}(0, I). \tag{5.16}$$

A soft update mechanism with a polyak averaging coefficient ρ is utilized to update the parameters of the target Q networks:

$$\phi_{\text{target,i}} = \rho\phi_{\text{target,i}} + (1-\rho)\phi_{\text{i}} \text{ for } i = 1, 2 \tag{5.17}$$

5.4 Quantum Policy Learning

This section first provides some computational foundations of quantum computing and VQC. Subsequently, it introduces the implementation framework at an overview level, followed by a detailed discussion on the circuit design and the design and gradient derivation of the loss function in VQC, providing the details of the implementation.

5.4.1 Quantum Computing Basics

The Dirac bra-ket notation provides a way to express vectors in a Hilbert space, which is the mathematical space of all possible quantum states. The column vector is denoted as $|\psi\rangle$, where ψ is the state of the quantum system. The corresponding row vector, which is the complex conjugate transpose (or adjoint) of $|\psi\rangle$, is denoted as $\langle\psi|$. The inner product between two states $|\psi\rangle$ and $|\phi\rangle$ is expressed as $\langle\psi|\phi\rangle$. An outer product yields an operator and is written as $|\psi\rangle\langle\phi|$.

A quantum system, such as a quantum bit or qubit, can be in a superposition of both its basis states simultaneously. Only upon measurement does the system collapse to one of the basis states with a certain probability determined by the coefficients of the superposition.

Mathematically, the state of a qubit in superposition can be represented as:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle \tag{5.18}$$

where $|\psi\rangle$ is the quantum state of the qubit, $|0\rangle$ and $|1\rangle$ are the basis states, and α and β are complex coefficients. The squared magnitudes $|\alpha|^2$ and $|\beta|^2$ give the probabilities of measuring the qubit in state $|0\rangle$ and $|1\rangle$ respectively, and they must satisfy the normalization condition: $|\alpha|^2 + |\beta|^2 = 1$.

Quantum entanglement describes a situation wherein the states of two or more qubits become correlated in such a way that the state of one qubit cannot be described independently of the state of the other qubit. An example of an entangled state is the Bell state. For a pair of qubits, the Bell states can be represented as:

$$|\Phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle) \tag{5.19}$$

$$|\Phi^-\rangle = \frac{1}{\sqrt{2}}(|00\rangle - |11\rangle) \tag{5.20}$$

5.4.2 Variational Quantum Circuits

VQCs are foundational components in contemporary quantum computing research. The core principle of VQCs lies in the hybrid quantum-classical optimization of parameterized quantum circuits to solve particular tasks. A VQC can be represented by a parameterized unitary $U(\boldsymbol{\omega}, \boldsymbol{\varphi})$ with adjustable parameters $\boldsymbol{\omega}$ and static parameters $\boldsymbol{\varphi}$. In the realm of quantum-classical optimization, optimization is used to fine-tune the $\boldsymbol{\omega}$ parameters, thereby directing the quantum circuit towards a desired computational result.

Similar to classical neural networks, VQCs can be constructed in a layered fashion. Each layer in a VQC consists of a series of quantum gates with their associated parameterized unitaries. By stacking multiple such layers, the expressiveness and computational power of VQCs can be enhanced, allowing it to capture more intricate quantum correlations. In a more compact notation, the state after K layers can be expressed as

$$|\psi_K\rangle = U_K(\boldsymbol{\omega}_K, \boldsymbol{\varphi}_K) \dots U_2(\boldsymbol{\omega}_2, \boldsymbol{\varphi}_2) U_1(\boldsymbol{\omega}_1, \boldsymbol{\varphi}_1)|\psi_0\rangle.$$

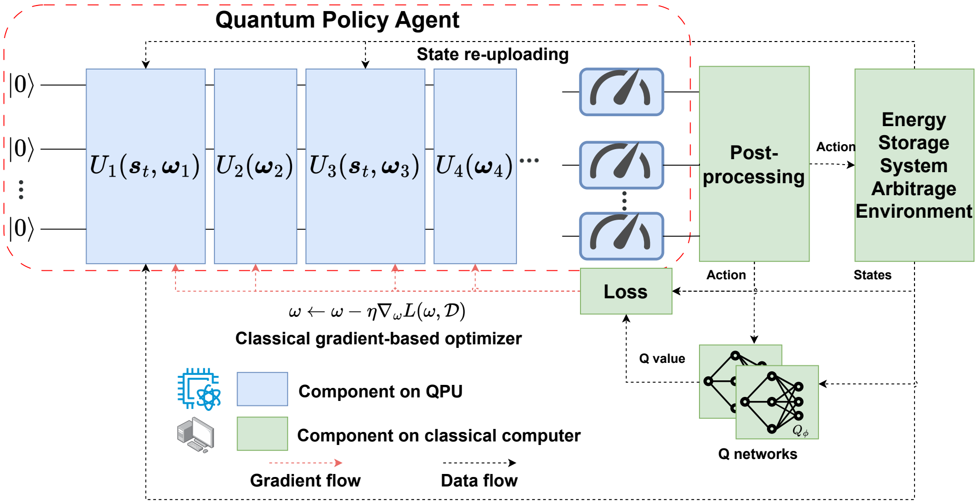
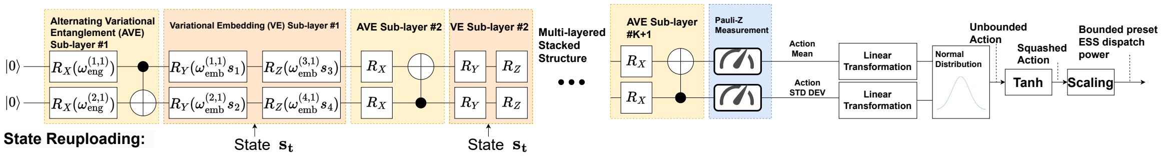

Quantum Policy Agent
State re-uploading
$|0\rangle$
$|0\rangle$
$|0\rangle$
$U_1(\boldsymbol{s}_t, \boldsymbol{\omega}_1)$
$U_2(\boldsymbol{\omega}_2)$
$U_3(\boldsymbol{s}_t, \boldsymbol{\omega}_3)$
$U_4(\boldsymbol{\omega}_4)$
Post-processing
Action
Energy Storage System Arbitrage Environment
Action
States
Loss
$\omega \leftarrow \omega - \eta \nabla_\omega L(\omega, \mathcal{D})$
Classical gradient-based optimizer
Q value
Q_ϕ
Q networks
Component on QPU
Component on classical computer
Gradient flow
Data flow

Figure 5.1: Framework of quantum policy learning.

5.4.3 Implementation Framework of Quantum Policy Learning

On the top of distinct features of quantum computing and VQCs, Fig.5.1 shows the framework of the proposed QPL for the EA of ESSs. The entire framework utilizes a quantum-classical hybrid structure, where the blue section represents the circuit deployable on the QPU, also known as the QPL agent. The green section signifies the classical component, primarily engaging in interactions with the environment and utilizes feedback data to construct a loss function, subsequently updating the parameters of the QPL agent through an optimizer. The state from the classical component is integrated into the QPL agent through methods of angle embedding and re-uploading. The QPL agent with the latest parameterized VQCs outputs discharge and charge decisions via quantum measurement.

In Fig.5.1, the green section is utilized during the QPL training phase but is not required during the testing phase. It is worth highlighting that when there are few qubits in VQCs, classical computers can simulate the qubit system, thereby allowing the leveraging of readily available classical computer simulations to conduct training. In the testing phase, upon securing a set of optimal VQC parameters, these are incorporated into the VQCs and deployed online on quantum processing units (QPUs). This approach is employed in the case studies in this paper, considering the high computational resource intensity of gathering environmental



Alternating Variational Entanglement (AVE) Sub-layer #1
Variational Embedding (VE) Sub-layer #1
AVE Sub-layer #2
VE Sub-layer #2
Multi-layered Stacked Structure
AVE Sub-layer #K+1
Pauli-Z Measurement
$|0\rangle$
$|0\rangle$
$R_X(\omega_{\text{eng}}^{(1,1)})$
$R_X(\omega_{\text{eng}}^{(2,1)})$
$R_Y(\omega_{\text{emb}}^{(1,1)}s_1)$
$R_Y(\omega_{\text{emb}}^{(2,1)}s_2)$
$R_Z(\omega_{\text{emb}}^{(3,1)}s_3)$
$R_Z(\omega_{\text{emb}}^{(4,1)}s_4)$
R_X
R_X
R_Y
R_Z
R_Y
R_Z
R_X
R_X
State Reuploading:
State $\mathbf{s_t}$
State $\mathbf{s_t}$
Action Mean
Action STD DEV
Linear Transformation
Linear Transformation
Normal Distribution
Unbounded Action
Squashed Action
Tanh
Bounded preset ESS dispatch power
Scaling

Figure 5.2: An example of the proposed quantum policy: a two qubits variational quantum circuit followed by the ensuing action handling strategies.

interactions.

5.4.4 Circuit Design for Quantum Policy

The paper devises a multi-layered stacked structure for the VQC utilized in the QPL agent, as shown in Fig. 5.2. K and k are used to denote the number of layers adopted in the VQC and the index of the layer. Each layer is segmented into two sub-layers: the alternating variational entanglement (AVE) sub-layer and the variational embedding (VE) sub-layer.

The primary ingredients of the AVE sub-layer are the parameterized x-rotation gates R_x and Controlled NOT (CNOT) gates.

R_x rotates a qubit around the X axis of the Bloch sphere by a specified angle. This rotation is represented by the operator: $R_x(\alpha) = e^{-i\alpha\sigma_{\mathbf{x}}/2}$, where α is the angle of rotation, and $\sigma_{\mathbf{x}}$ is the Pauli-X matrix given by: $\sigma_{\mathbf{x}} = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$

Utilizing the matrix exponential and the properties of the Pauli-X matrix, the above expression is expanded as:

$$\begin{aligned} e^{-i\alpha\sigma_{\mathbf{x}}/2} &= I\cos(\alpha/2) - i\sigma_{\mathbf{x}}\sin(\alpha/2) \\ &= \begin{bmatrix} \cos(\alpha/2) & -i\sin(\alpha/2) \\ -i\sin(\alpha/2) & \cos(\alpha/2) \end{bmatrix} \end{aligned} \tag{5.21}$$

In the AVE sub-layer involving a N qubits system, each qubit begins with a R_x gate. This operation can be represented by the following unitary operator:

$$U_{\text{eng}}^{(k)}\left(\omega_{\text{eng}}^{(:,k)}\right) = \bigotimes_{j=1}^{N} e^{-i\omega_{\text{eng}}^{(j,k)}\sigma_{\mathbf{x}}/2} \tag{5.22}$$

where $\bigotimes$ represents the tensor product; j is the index for qubits; $\omega_{\text{eng}}^{(:,k)}$ is a vector of learnable rotation angles for the R_x gates in the k-th AVE sub-layer.

After a series of R_x, using a ring structure of CNOT gates between consecutive qubits with alternating directions to achieve entanglement provides the potential for quantum algorithms to outperform their classical counterparts.

If k is odd, each qubit $j-1$ acts as the control for the following qubit j:

$$U_{\text{CNOT}}^{(k)} = \prod_{j=1}^{N} I^{\otimes(j-2)} \otimes \text{CNOT}_{j-1,j} \otimes I^{\otimes(N-j)}. \tag{5.23}$$

If k is even, the direction of control is reversed, thereby implementing alternating manner to facilitate the exploration in Hilbert space:

$$U_{\text{CNOT}}^{(k)} = \prod_{j=1}^{N} I^{\otimes(j-2)} \otimes \overline{\text{CNOT}}_{j,j-1} \otimes I^{\otimes(N-j)}. \tag{5.24}$$

Here, the terms $I^{\otimes(j-2)}$ and $I^{\otimes(N-j)}$ denote tensor products of identity matrices. Their role is to preserve the state of qubits not directly involved in the CNOT operation. The notation $\text{CNOT}_{j-1,j}$ signifies the CNOT gate where the qubit $j-1$ serves as the control and qubit j as the target. Conversely, $\overline{\text{CNOT}}_{j,j-1}$ indicates a reversed CNOT gate with qubit j as the control and qubit $j-1$ as the target.

The matrix representations for these gates are given by: $\text{CNOT} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix}$ and $\overline{\text{CNOT}} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix}$.

Following the AVE sub-layer is the VE layer. In this layer, the state vectors of EA tasks influence quantum state variables in the Hilbert space through angle embedding.

Ref. [143] proposes embedding each element of the state vector through the R_x gate of each qubit, with the number of dimensions corresponding to the number of qubits. However, considering the dimensions of the state vectors for real-world tasks are often large, this approach is not yet practical in the NISQ era due to the availability of qubits and their noise resistance capabilities. To overcome this challenge, a structure of alternating stacking R_y and R_z gates within a single qubit to embed the high-dimensional state vector into the VE sub-layer is devised instead. Alternating between R_y and R_z gates enhance the expressibility of the VQCs, allowing for a richer set of quantum states and forming a foundation for universal quantum computation.

As demonstrated in the example circuit in Fig. 5.2, the initial step involves determining the requisite number of qubits, denoted as N. Subsequently, the initial N components of the state vector are mapped into the R_y gate. This is followed by the embedding of the subsequent N components into the R_z gate. This alternating embedding process continues in a similar fashion until the last component in the state vector.

The rotation angle range for R_y and R_z gates is typically from 0 to 2π. To achieve a smoother feature embedding, rather than directly using the normalized state as the angle, we introduce a set of variational parameters $\omega_{\text{emb}}^{(:,k)}$ and consider the product of this parameter with the feature as the embedding angle.

Assuming the dimension of the state vector, $|S|$, is greater than N and is equal to $2N$, the unitary operator for VE can be expressed as:

$$U_{\text{emb}}^{(k)}\left(\omega_{\text{emb}}^{(:,k)}\right) = \bigotimes_{j=1}^{N} e^{-i\omega_{\text{emb}}^{(N+j,k)} * s_{N+j}\sigma_{\mathbf{z}}/2} \bigotimes_{j=1}^{N} e^{-i\omega_{\text{emb}}^{(j,k)} * s_j \sigma_{\mathbf{y}}/2} \tag{5.25}$$

where $\sigma_{\mathbf{y}}$ and $\sigma_{\mathbf{z}}$ are the Pauli-Y and Pauli-Z matrices given by $\sigma_{\mathbf{y}} = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}$ and $\sigma_{\mathbf{z}} = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$, respectively.

After K layers of AVE and VE sub-layers, one layer of AVE sub-layer is added before the final output of the VQC result. Its corresponding unitary operator is

$$U_{\text{CNOT}}^{(K+1)} U_{\text{eng}}^{(K+1)}\left(\omega_{\text{eng}}^{(:,K+1)}\right) = U_{\text{CNOT}}^{(K+1)} \bigotimes_{j=1}^{N} e^{-i\omega_{\text{eng}}^{(j,K+1)}\sigma_{\mathbf{x}}/2}. \tag{5.26}$$

5.4.5 Measurement

Quantum circuits typically conclude with measurement gates, serving to convey information to the external environment and reveal information about the associated quantum state. The action dimension of the task in this paper is one, but there are two parameters modeling this action distribution: the mean and the standard deviation. Therefore, two qubits need to be chosen for measurement. Without loss of generality, this paper selects the first and second qubits (i.e., qubit 0 and qubit 1) and uses the Pauli-Z gate as the required Hermitian operator for measurement. A similar structure can be adapted to generalize to the action dimension larger than one if needed.

The Pauli-Z matrix, when used as a measurement operator, has distinct eigenvalues $\lambda = \pm 1$. These eigenvalues correspond to the possible outcomes of a quantum measurement on a computational basis. The eigendecomposition of the Pauli-Z matrix is given by:

$$\sigma_z = |0\rangle\langle 0| - |1\rangle\langle 1| \tag{5.27}$$

Specifically, the state $|0\rangle$ is associated with the eigenvalue $+1$, while the state $|1\rangle$ corresponds to -1. Performing the Pauli-Z measurement of qubit 0 causes the quantum state to collapse to the basis state characterized by the eigenvector of the Pauli-Z operator. When the state collapses to $|0\rangle$, the measurement output is $+1$; when it collapses to $|1\rangle$, the measurement output is -1.

For a two-qubit VQC as an example, suppose the arbitrary quantum state after multiple AVE and VE layers is $|\psi'\rangle$, then the expected output of the Pauli-Z measurement on qubit 0 is

$$\langle \sigma_{z,\omega}^{(0)} \rangle \ = \langle \psi' | \sigma_z \otimes I | \psi' \rangle, \tag{5.28}$$

where the superscript (0) denotes the index of the qubit, and the subscript ω indicates the expected output is dependent on the parameter set $\boldsymbol{\omega}$ of the VQC. Likewise, $\langle \sigma_{z,\omega}^{(1)} \rangle = \langle \psi' | I \otimes \sigma_z | \psi' \rangle$. Due to Born's rule, the value of $\langle \sigma_z \rangle$ is a real number in the range [-1,1].

Examining Eqs. (5.22) to (5.28), the entire derivation process for $\langle \sigma_{z,\omega}^{(0)} \rangle$ is differentiable. Leveraging this feature, one can update the parameter set $\boldsymbol{\omega}$ of the VQC based on the gradient information of $\langle \sigma_{z,\omega}^{(0)} \rangle$ with respect to $\boldsymbol{\omega}$, thereby accomplishing the computational task.

In a physically-implemented VQC, due to the indeterminacy of the quantum state before measurement, the parameter-shift rule can be employed to approximate the gradient information. By using additional two circuit evaluations, one can obtain the gradient of an observable (e.g., $\langle\sigma_{z,\omega}^{(0)}\rangle$) with respect to an arbitrary adjustable parameter ω_i.

$$\begin{aligned}\frac{d\langle\sigma_{z,\omega}^{(0)}\rangle}{d\omega_i} = \frac{1}{2}\big(&\langle\psi'(\omega_i+\pi/2)|\sigma_{z,\omega}^{(0)}\otimes I|\psi'(\omega_i+\pi/2)\rangle - \\ &\langle\psi'(\omega_i-\pi/2)|\sigma_{z,\omega}^{(0)}\otimes I|\psi'(\omega_i-\pi/2)\rangle\big)\end{aligned} \tag{5.29}$$

If a VQC is implemented using a classical simulator, one can access the intermediate states of the VQC. Therefore, classical automatic differentiation techniques can be leveraged to derive the analytical expression for the gradient. For example, to calculate the gradient of $\sigma_{z,\omega}^{(0)}$ w.r.t $\omega_{\text{eng}}^{(1,K+1)}$ (i.e., the parameter on the qubit 0 of the last AVE layer, denoted as w_1 below for simplicity), we can differentiate $\sigma_{z,\omega}^{(0)}$:

$$\begin{aligned}\frac{d\sigma_{z,\omega}^{(0)}}{dw_1} &= \frac{d}{dw_1}\langle\psi'|(Z\otimes I)|\psi'\rangle \\ &= \left\langle\frac{d\psi'}{dw_1}|(Z\otimes I)|\psi'\right\rangle + \left\langle\psi'|(Z\otimes I)|\frac{d\psi'}{dw_1}\right\rangle\end{aligned} \tag{5.30}$$

Assuming the quantum state before passing through this AVE layer is ψ_K, $|\frac{d\psi'}{dw_1}\rangle$ can be determined by

$$|\frac{d\psi'}{dw_1}\rangle = \overline{\text{CNOT}}\left(R_x(w_2)\otimes\frac{dR_x(w_1)}{dw_1}\right)|\psi_K\rangle, \tag{5.31}$$

where $\frac{dR_x(w_1)}{dw_1} = \begin{bmatrix} -\frac{\sin(w_1/2)}{2} & -\frac{i\cos(w_1/2)}{2} \\ -\frac{i\cos(w_1/2)}{2} & -\frac{\sin(w_1/2)}{2}\end{bmatrix}$. $\langle\frac{d\psi'}{dw_1}|$ is the Hermitian transpose of $|\frac{d\psi'}{dw_1}\rangle$.

5.4.6 Quantum Policy

This subsection is dedicated to extending the policy loss function in Eq. (5.15) and its corresponding gradient update strategy to VQC, thus enabling effective update of the parameters of the QPL agent. In Eq. (5.15), Q_{ϕ_j} remains in its classical form. The quantum versions of the reparameterized action term $\tilde{a}(s,\xi)$ and the log-likelihood term $\log\pi(\tilde{a}(s,\xi)\mid s)$

related to the policy network need to be derived. The derivations and analyses in this section are based on one-dimensional actions, but the relevant conclusions can also be easily extended to multi-dimensional cases.

As shown in Eq. (5.16), classic SAC employs a squashed Gaussian policy, in which a factored Gaussian action needs to go through a tanh function. The QPL agent also adopts a similar squashing strategy.

The expected value of the measurement on qubit 0, denoted as $\langle\sigma_{z,\omega}^{(0)}\rangle$, is linearly transformed to span the range [-3,3], allowing it to cover the primary effective region of the tanh function. Then, the transformed value is treated as the expected value of the action distribution: $\mu_\omega := 3\langle\sigma_{z,\omega}^{(0)}\rangle$, where the subscript ω indicates that μ_ω is dependent on the VQC parameters ω. Likewise, the expected value of the measurement on qubit 1 is linearly transformed to span the range [0,5] to represent the standard deviation of the action distribution $\sigma_\omega := \frac{5}{2}(\langle\sigma_{z,\omega}^{(1)}\rangle + 1)$.

μ_ω and σ_ω jointly determine the distribution of the vanilla action u. Directly sampling vanilla actions from the policy might prevent the gradient from propagating from the output of the policy to its parameters. The reparametrization trick is needed to make the action a deterministic function of the policy parameters ω and noise ξ. Hence, the vanilla action can be written as:

$$u_\omega(s,\xi) = \left(3\langle\sigma_{z,\omega}^{(0)}(s)\rangle + \frac{5}{2}(\langle\sigma_{z,\omega}^{(1)}(s)\rangle + 1) \odot \xi\right), \tag{5.32}$$

where $\xi \sim \mathcal{N}(0, I)$.

The support of the distribution of u is (-inf, inf). Define the probability density function (p.d.f) of this distribution as $f_\omega(u|s)$.

A squashed action $\tilde{a}$ can be obtained after the tanh transformation:

$$\tilde{a}_\omega(s,\xi) = \tanh(u_\omega(s,\xi)), \tag{5.33}$$

with the support transitioned to [-1, 1].

The p.d.f of the squashed action is:

$$\pi_\omega(\tilde{a} \mid s) = f_\omega(u_\omega|s) \cdot |1 - \tanh^2(u_\omega)|^{-1}. \tag{5.34}$$

Further, the log-likelihood for the squashed action can be derived:

$$\log \pi_\omega(\tilde{a} \mid s) = \log f_\omega(u_\omega|s) - \log\left(1 - \tanh^2(u_\omega)\right) \tag{5.35}$$

Since u_ω follows a normal distribution, the first term of Eq. (5.35), i.e., the log-likelihood of the normal distribution, is

$$\begin{aligned}\log f_\omega(u_\omega|s) &= -\log(\sigma_\omega) - \log(\sqrt{2\pi}) - \frac{1}{2}\left(\frac{u_\omega - \mu_\omega}{\sigma_\omega}\right)^2 \\ &= -\log\left(\frac{5\sqrt{2\pi}}{2}\right) - \log(\langle\sigma_{z,\omega}^{(1)}\rangle + 1) - 2\left(\frac{u_\omega - 3\langle\sigma_{z,\omega}^{(0)}\rangle}{5(\langle\sigma_{z,\omega}^{(1)}\rangle + 1)}\right)^2\end{aligned} \tag{5.36}$$

Based on

$$\begin{aligned}1 - \tanh^2(u_\omega) &= \frac{(e^{u_\omega} + e^{-u_\omega})^2 - (e^{u_\omega} - e^{-u_\omega})^2}{(e^{u_\omega} + e^{-u_\omega})^2} \\ &= \frac{4e^{-2u_\omega}}{(1 + e^{-2u_\omega})^2},\end{aligned}$$

the closed-form expression for the second term of Eq. (5.35) is:

$$\begin{aligned}&\log\left(1 - \tanh^2(u_\omega)\right) \\ &= \log(4) + \log\left(e^{-2u_\omega}\right) - \log\left(\left(1 + e^{-2u_\omega}\right)^2\right) \\ &= 2\left(\log(2) - u_\omega - \log\left(1 + e^{-2u_\omega}\right)\right).\end{aligned} \tag{5.37}$$

Combining Eqs (5.32) through (5.37), the loss function for training the QPL agent is

$$L(\omega, \mathcal{D}) = \underset{\substack{s \sim D \\ \xi \sim \mathcal{N}}}{\mathrm{E}}\left[\vartheta \log \pi_\omega\left(\tilde{a}_\omega(s, \xi) \mid s\right) - \min_{j=1,2} Q_{\phi_j}\left(s, \tilde{a}_\omega(s, \xi)\right)\right]. \tag{5.38}$$

Using the pathwise gradient estimation introduced in [148], an unbiased gradient of (5.38) can be approximated by

$$\begin{aligned}\nabla_\omega L(\omega, \mathcal{D}) = \vartheta \nabla_\omega \log\left(\pi_\omega\left(\tilde{a}_t \mid s_t\right)\right) + \\ \nabla_\omega \tilde{a}_t\left(\vartheta \nabla_{\tilde{a}_t} \log\left(\pi_\phi\left(\tilde{a}_t \mid s_t\right)\right) - \nabla_{\tilde{a}_t} Q\left(s_t, \tilde{a}_t\right)\right).\end{aligned} \tag{5.39}$$

Here, ϑ is treated as a constant; $\nabla_\omega \log\left(\pi_\omega\left(\tilde{a}_t \mid s_t\right)\right)$ is jointly determined by (5.29), (5.32), (5.35) to (5.37); $\nabla_{\omega} \tilde{a}_t$ is jointly determined by (5.29), (5.32), and (5.33); $\nabla_{\tilde{a}_t} \log\left(\pi_\phi\left(\tilde{a}_t \mid s_t\right)\right)$ is jointly determined by (5.35) to (5.37), and $\tilde{a} = \tanh^{-1}(u)$; $\nabla_{\tilde{a}_t} Q\left(s_t, \tilde{a}_t\right)$ is obtained via the auto differentiation in a DL framework.

The procedure to implement the QPL is summarized in Algorithm 7.

Algorithm 7 Quantum Policy Learning

```
Input: VQC parameters $\boldsymbol{\omega}$ initialized by a uniform distribution within a $[0, 2\pi]$ interval,
  Q-function parameters $\phi_1$ and $\phi_2$ initialized by He Normal, empty replay buffer $D$
Set target parameters equal to main parameters $\phi_{\text{target},1} = \phi_1, \phi_{\text{target},2} = \phi_2$
repeat
  Observe state $s$, sample a vanilla action $u$ with (5.32), obtain squashed action $\tilde{a}$ with (5.33).
  Execute $\tilde{a}$ in the environment
  Observe next state $s'$, reward $r$, and terminal signal $d$
  Store $(s, \tilde{a}, r, s', d)$ in replay buffer $D$
  if $s'$ is terminal, then
    reset environment.
  end if
  if it's time to update then
    for $j$ in range(preset update times) do
      Randomly sample a batch of transitions $\{(s, \tilde{a}, r, s', d)\}$ from $D$.
      Compute targets for the Q functions with (5.14).
      Update Q-functions by one step of gradient descent using (5.13).
      Update quantum policy with quantum policy gradient in (5.39).
      Update target networks with (5.17).
    end for
  end if
until convergence
```

5.5 Numerical Results

In this section, the viability and superiority of the proposed QPL algorithm are examined, alongside a discussion on the factors influencing the performance of VQCs. The data spanning one year has been retrieved from a specific location within the PJM real-time energy market. This dataset comprises 365 data entries, each featuring 24 dimensions corresponding to the individual hours within a day. 80% (292 days) of the dataset is used for training, and the rest (73 days) is used for testing. Table 5.1 lists the parameters of the ESS. Relevant hyperparameters of the QPL algorithm used in the following case studies are shown in Table 5.2.

The computational setup used in this study involves both classical and quantum computers to implement a hybrid computational framework. The conventional neural network aspects are developed using PyTorch. Pennylane and IBM's Open Quantum Assembly Language (QASM) are utilized for the construction and simulation of the VQC. Training of the VQCs is carried out on processors equipped with AMD EPYC 7763 running at a frequency of 2.45 GHz. The testing phase is conducted using a combination of the aforementioned processors and cloud-based resources: the ibmq_qasm_simulator simulator and the ibm_lagos QPU.

Parameter	Value	Parameter	Value	Parameter	Value
U	30 MWh	η_{ch}	0.9	η_{dis}	0.9
δ_h	1 hour	$\overline{SOC}$	0.9	$\underline{SOC}$	0.1
$P_{\max}$	5 MW	c	10 \$/MW		

Table 5.1: Parameters of Energy Storage System.

Parameter	Value	Parameter	Value
replay buffer size	1e4	exploration steps	2e3
batch size	32	Q function hidden layer size	[64,64]
ρ	0.995	γ/ϑ	0.99/0.1
Optimizer	Adam	quantum policy/ Q function learning rate	1e-3/3e-3

Table 5.2: Hyperparameters of the QPL algorithm.

5.5.1 Quantum Advantage: Expressivity

To validate the practicality and superiority of the proposed QPL, the training results of the proposed QPL algorithm, characterized by a policy driven by pure quantum circuits, are compared with those of the classical SAC. The moving average of episodic returns on the training set, i.e., $R = \sum_{t=0}^{23} \gamma^t r_t$, with a rolling window of 200, is utilized as a performance metric to gauge the training efficacy. Given the fluctuations in electricity prices, both intra-day and inter-day, this metric can serve as the expected performance of the RL agent under varying electricity prices within the training dataset.

Fig. 5.3 showcases the evolution of episodic returns throughout the training process for three types of RL agents: QPL (20 paras), Classic SAC (20 paras), and Classic SAC (1280 paras), in which the shadowed areas represent the standard deviation of the performance over three random seeds. In the case studies, unless otherwise specified, the QPL adopts a three-layer structure with two qubits, requiring only 20 variational parameters to be trained. In contrast, one of the comparative counterparts, the Classic SAC (1280 paras), encompasses a policy network with two hidden layers, each with 32 neurons, representing a more common and practical configuration in the realm of DRL. The Classic SAC (20 paras) mirrors the parameter volume adopted in the QPL. The performance of QPL markedly surpasses that of Classic SAC (20 paras): while utilizing the same number of trainable parameters, 20 in this case, QPL manages to converge more swiftly and stably to an episodic return of 15.69 with a standard deviation of 0.26, improving over 60% compared to the 9.73 achieved by the Classic SAC (20 paras). QPL also outperforms Classic SAC (1280 paras). QPL showcases an almost monotonically increasing trend from the beginning of training, converging to around 15.69 in merely about 40k steps, whereas the Classic SAC (1280 paras) requires approximately 60k steps. This suggests that the QPL agent exhibits a higher utilization rate of existing experiences while not compromising its exploratory nature. This is evidenced by a relative difference of less than 1% in the converged values compared to the 15.76 attained by the Classic SAC (1280 paras).

Utilizing only about 1.6% of the parameter quantity, QPL is comparable with the prevalent classic policy networks. This quantum advantage is associated with the trainability and expressibility conferred by the VQCs devised in this study [141, 149]. The empirical result

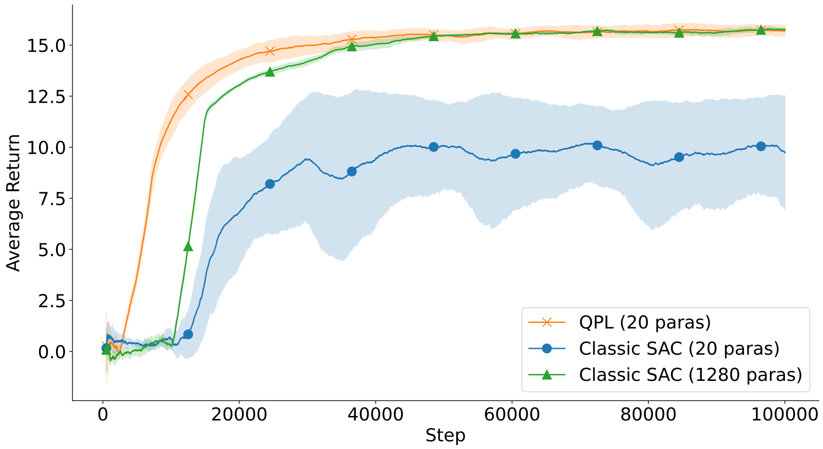


Figure 5.3: The impact of varying layer quantities in the quantum circuit on the training performance of the proposed QSAC agent.

demonstrates the proposed QPL can approximate the solution space between the state and action in Hilbert space sufficiently and efficiently.

5.5.2 Implantation on Real Quantum Hardware

Utilizing the trained parameters to build the quantum circuit enables the quantum circuit operating on the QPU to efficiently manage the ESS in an online manner. In this section, we refer to this phase as the testing phase. To demonstrate the generalization ability of the trained QPL agent and its resilience to actual decoherence noise, a 24-hour period is selected from the test set for the RL agents to schedule, which is depicted by a black curve in Fig. 5.4.

Two approaches are adopted here as benchmarks. One assumes that the 24-hour electricity price information is known and noise-free, based on which an optimization model can be constructed to derive a scheduling scheme. This method is denoted as Opti-Complete. In reality, this method is impractical because assuming known future electricity prices in the real-time market is unrealistic. The other method is Model Predictive Control (MPC), a more practical approach designed for online operation. Employing a rolling scheduling strategy, it utilizes the noisy electricity price predictions available over a limited future time horizon. Only the scheduling decision for the next time step will be deployed. In this ex-

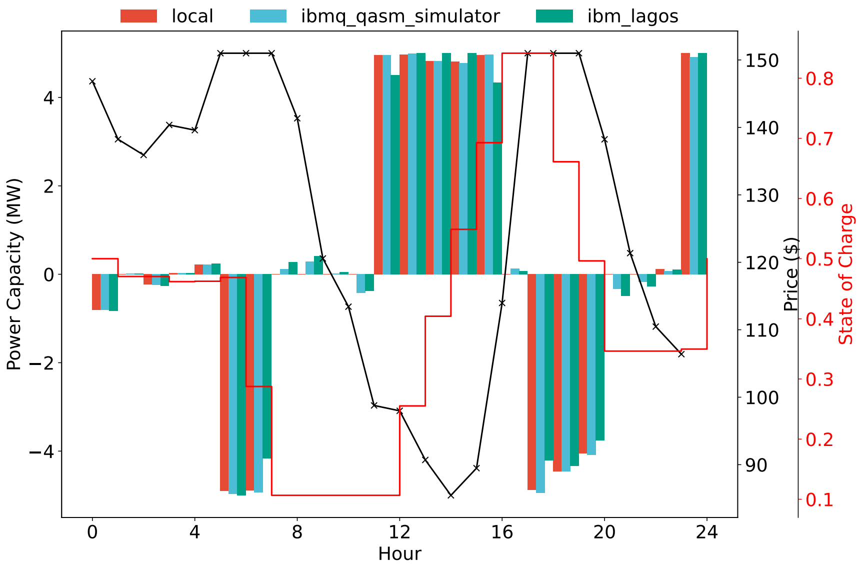


Figure 5.4: Power scheduling results of a trained agent on a local exact simulator, a cloud measurement-based simulator, and a real quantum hardware.

ample, the prediction error is modeled using a Gaussian distribution with a mean of zero. The standard deviation of the prediction error starts at 0.02 and increases linearly as the distance between the future time steps and the current time step expands. The comparison criterion is the total amount of net profit gained by the ESS within this 24-hour period. Opti-Complete can earn \$293.33, which can be regarded as the upper bound of the profit. With noisy price prediction, MPC can only achieve a profit of \$178.72. The scheduling results outputted by QPL are displayed in the bar charts in Fig. 5.4, which materializes a profit of \$260.72. For a specific electricity price scenario, optimization-based methods require one or several instances of modeling and solving optimization problems. RL approaches mitigate this challenge, enabling instant adaptation to multiple unknown scenarios through a single round of offline training. Moreover, the results here empirically demonstrate that the QPL agent can outperform the MPC method in terms of profitability, which is close to the theoretical optimal value.

The trained circuit was tested online on three different platforms: a local simulator, the ibmq_qasm_simulator, and the ibm_lagos. Pennylane's default qubit is a simulator that can generate theoretically exact expected outcomes from quantum measurements, serving as an ideal benchmark for quantum computations. The ibmq_qasm_simulator, a cloud-based entity, replicates the stochastic nature of quantum measurements with a count of 10,000, offering a closer approximation to real quantum mechanics. The ibm_lagos is a superconducting quantum hardware with seven qubits, enabling the genuine testing and evaluation of circuits in a real-world quantum computing environment, showcasing the practicability and limitations primarily due to decoherence and imperfect quantum gates in the NISQ era. Fig. 5.5 presents the specifications and configurations of the ibm_lagos QPU. In addition to the topology of the qubits, the measurement (i.e., readout) errors, single-qubit gate errors, and two-qubit gate errors are visualized using a heat map. The qubits in the ibm_lagos are manipulated using microwave pulses to perform quantum computations.

The bar charts in Fig. 5.4 illustrate the scheduling outcomes generated by three different platforms. The primary finding is that, despite the presence of measurement errors and various hardware imperfections, such as gate errors and qubit decoherence, the scheduling outcomes across the three platforms are close. Using the scheduling results on the local simulator as a baseline, the L2 norm distances between the outcomes obtained with the ibmq_qasm_simulatorr and the ibm_lagos, relative to the baseline, are found to be 0.6678 and 1.5682, respectively. The insignificant deviation can be primarily attributed to the fact that the QPL employs a low-depth and low-width layer structure. This delicate design makes it less prone to errors, consequently enhancing its resilience in a real-world quantum computing environment. Furthermore, these results are quite promising, substantiating the feasibility of executing EA tasks through the proposed QPL in an online setting.

5.5.3 VQC structure evaluation

This part is dedicated to evaluating the impact of VQC design.

5.5.3.1 Number of layers

Fig. 5.6 depicts the curves illustrating the relationship between the optimization performance and the number of training iterations for the proposed VQCs with different layer counts. Apart from the VQCs with only one layer, the performance of the remaining VQCs is relatively similar. The VQCs with a single layer exhibit a substantial variance in returns,

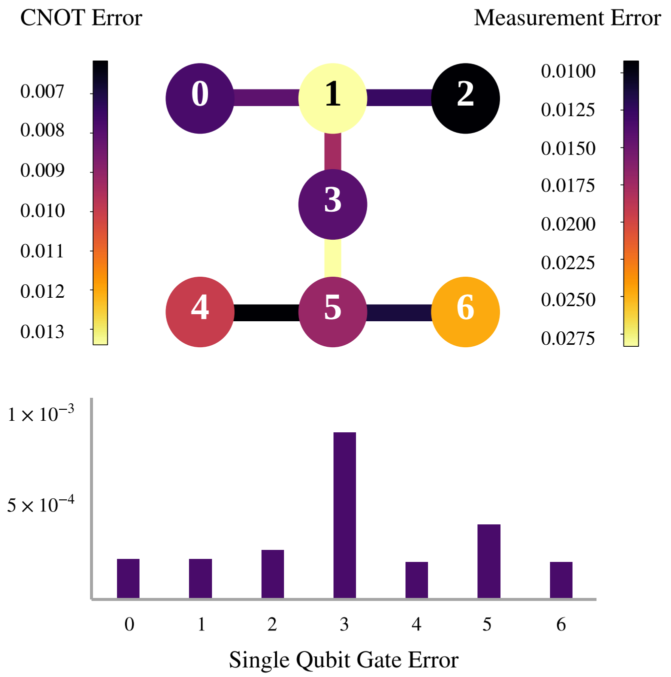


Figure 5.5: Specifications and configurations of the IBM-Lagos quantum processing unit

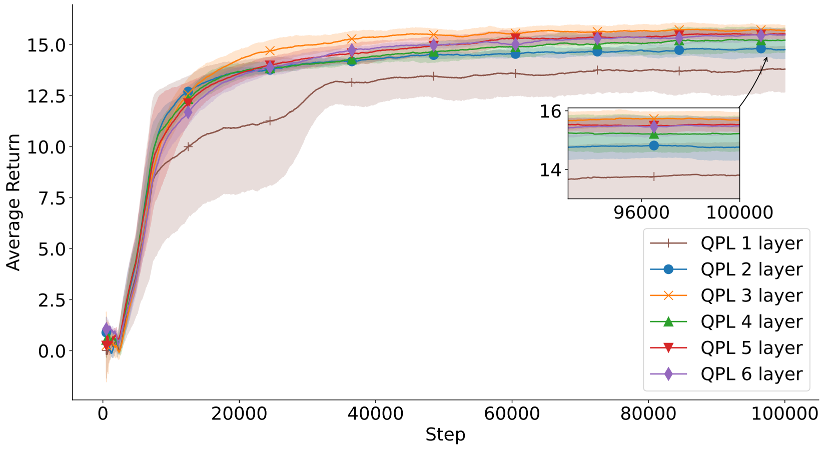


Figure 5.6: The impact of varying layer quantities in the proposed VQCs on the training performance of the proposed quantum policy agent.

and the policy performance it converges to is significantly inferior compared to the others. The heightened variance in performance exhibited by the one-layer VQCs is primarily due to their inability to implement state reuploading, resulting in a constrained learning capacity to generalize complex patterns, heightened sensitivity to training data fluctuations, and limited expressivity.

Table 5.3 takes a closer look at the values to which the curves in Fig. 5.6 eventually converge. It can be observed that the mean value exhibits a gradual increase until three layers, suggesting a potential improvement in model optimality. Beyond this point, a stabilization in mean values is observed, hinting at a saturation point in performance enhancement. The standard deviation values exhibit a downward trend, implying a potential improvement in result consistency as the layer number increases. Furthermore, the parameter count, which escalates linearly, signifies a consistent increment in the complexity of the quantum policy VQCs with the addition of layers.

Moreover, the polynomial parameter count $6 \cdot K + 2$ (where K represents the number of layers) directly corresponds to the complexity and scale of the quantum policy within VQCs. This growing parameter count not only mirrors the escalating complexity but also represents an expansion in the scale of the VQCs, influencing the quantum computational resources required. To strike a balance between achieving a robust performance and avoiding overcomplication, thereby fostering greater practicality in the NISQ era, a three-layer structure emerges as a viable choice. This preference is substantiated by the enhancement in mean values up until three layers, indicating a substantial gain in training performance. Moreover, the relatively low standard deviation at this point suggests satisfactory stability. Thus, opting for a three-layer structure could potentially encapsulate the desired optimality while retaining computational feasibility, paving the way for more practical applications in the evolving quantum landscape.

5.5.3.2 Circuit design

This subsection intends to compare the VQC layer structure presented in Fig. 5.2 with other layer structures, quantify their comparative performances, and distill some guidelines for VQC design oriented towards QPL.

Layer number	Mean Value	Standard Deviation	Parameter Count
1	13.81	1.12	8
2	14.76	0.45	14
3	15.69	0.26	20
4	15.23	0.59	26
5	15.53	0.25	32
6	15.48	0.28	38

Table 5.3: Post-training Performance of VQCs with Different Numbers of Layers

We refer to the VQC layer structure proposed in this paper as the QPL. The counterparts for comparison include the QAOA-inspired layer introduced in [150], the quantum-enhanced layer introduced in [151], the QPL without entanglement, and the conventional multi-layer stacked VQC. Specifically, Fig. 5.7(a) showcases a quantum-enhanced layer characterized by a sequential arrangement of Hadamard gates, R_z gates, and ZZ gates. The Hadamard gates are utilized to prepare the qubit in a quantum superposition state, whereas the ZZ gates, also known as Ising ZZ coupling gates, are employed to parameterize the circuit and introduce entanglement. The quantum-enhanced layer operates with three qubits, with seven trainable parameters in each layer. Fig. 5.7(b) depicts a QAOA-inspired layer characterized by the ZZ gates segregating the state embeddings. Fig. 5.7(c) corresponds to the exclusion of the CNOT gates in the proposed QPL, thereby eliminating the entanglement phenomena between the two qubits. Fig. 5.7(d) integrates state variables into the quantum state, utilizing stacked R_x gates and CNOT gates to formulate VQCs. Its paramount distinction from all the aforementioned circuits is the non-adoption of a state reuploading strategy.

In the subsequent experiments, the five types of VQCs depicted in Fig. 5.2 and Fig. 5.7 are employed independently as quantum policies. The training performance of the quantum policies in the EA environment, primarily assessed through average returns and stability, is utilized to evaluate the efficacy of these five VQCs in accomplishing EA tasks. It is noteworthy that, except for the multi-layer stacked circuit, which utilizes a four-layer setup, all other VQCs adopt a three-layer structure, thereby maintaining the number of trainable parameters at 20 or 21. This measure is taken to avoid any potential influence of the

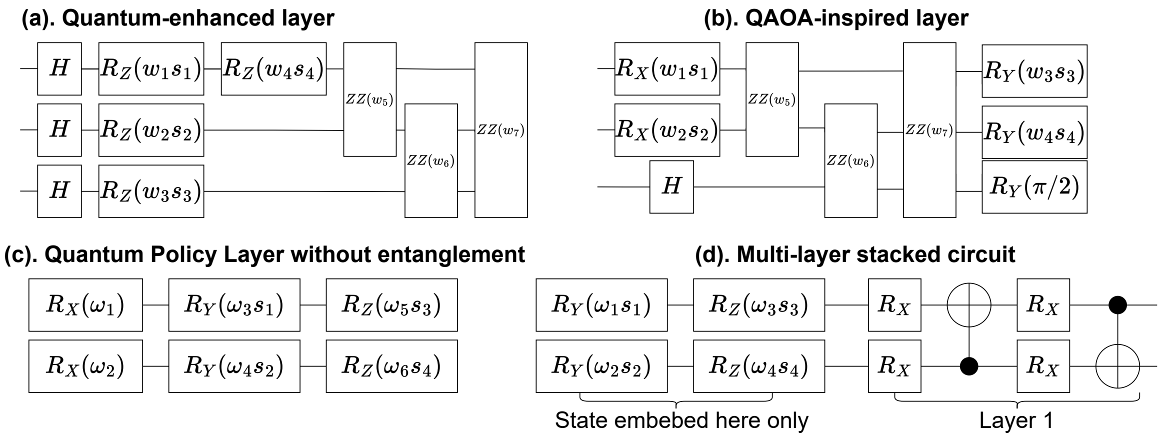


Figure 5.7: Four VQC layer structures serving as comparison counterparts.

parameter count on the experimental outcomes.

Fig. 5.8 depicts the effect of variations in circuit design on the training efficacy of the quantum policy agent. Table 5.4 focuses on the convergence points of the curves in Fig. 5.8, reporting the final performance of the VQCs across five different circuit structures. In Fig. 5.8, the proposed QPL structure evidently outperforms other circuit designs, both in terms of optimization efficacy and training stability. Following closely is the QPL structure that does not employ entanglement. The QAOA-inspired and quantum-enhanced circuits are slightly inferior to the preceding two. The least effective is the VQC, which does not implement a state reuploading scheme. It is observed that the QAOA-inspired and quantum-enhanced circuits, along with the VQC without a state reuploading scheme, exhibit a larger variance during the training process. In contrast, the circuits utilizing the proposed QPL structure do not have this issue.

In comparison to the quantum-enhanced and QAOA-inspired structure, the proposed QPL stands out with an impressive increment of roughly 16.9% and 14.7% in the mean value, respectively, showcasing a significant stride in optimizing performance metrics. This superiority can be attributed to the rational arrangement of AVE and VE sub-layers within the QPL structure, coupled with the sequential utilization and arrangement of R_x, R_y, and R_z gates. This design ensures that the quantum states can be amply dispersed across

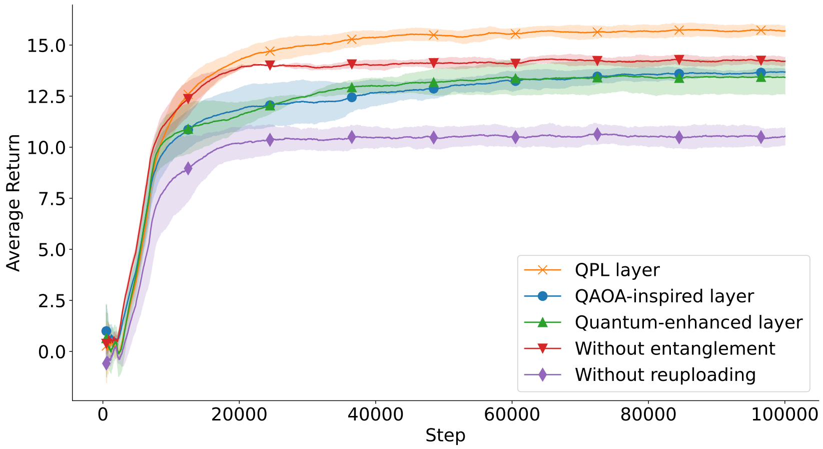


Figure 5.8: The impact of varying circuit design on the training perforofnce of the proposed quantum policy agent.

the Hilbert space while effectively integrating the environmental information provided by the state vector. When viewed against the QPL without entanglement, the QPL structure exhibits a 10.4% escalation in mean value. This result substantiates that the incorporation of inter-qubit entanglement interactions grants the QPL a significant edge by fostering complex correlations and richer computational landscapes. The QPL overshadows the VQC without a state reuploading scheme by a massive 49%, which necessitates the adoption of a state reuploading scheme in the QPL algorithm. The state reuploading scheme augments the expressive capability of the quantum circuit, thereby bolstering its training performance.

VQC name	Mean Value	Standard Deviation
QPL	15.69	0.26
QAOA-inspired [150]	13.68	0.15
quantum-enhanced [151, 152]	13.42	0.82
QPL w/o entanglement	14.21	0.20
VQC w/o state reuploading	10.53	0.42

Table 5.4: Post-training Perforofnce of VQCs with Different Layer Designs

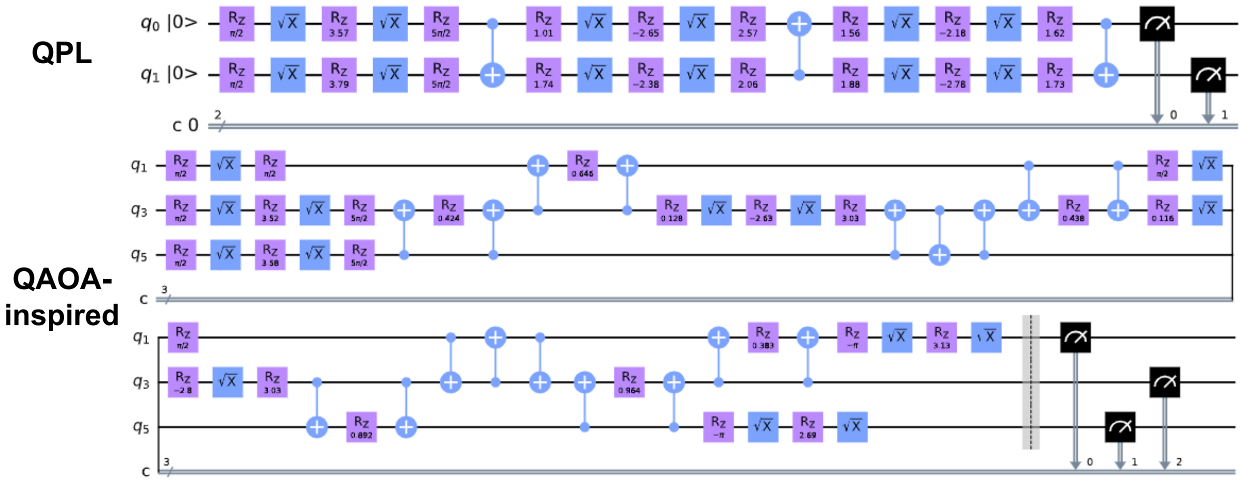


Figure 5.9: quantum circuits implemented at ibm_lagos after compilation: QPL v.s. QAOA-inspired.

A notable advantage of QPL is its low depth and narrow width, which become even more pronounced when compiled into circuits compatible with real quantum hardware. Fig. 5.9 shows the quantum circuits with QPL and QAOA-inspired layers executed on the ibm_lagos hardware following compilation. It is evident that the complexity of the QPL after compilation is significantly less than that of the QAOA-inspired. IBM-Lagos supports $rz(R_z)$, $sx(\sqrt{X})$, and cx(CNOT) gates.

The quantum circuit specifications after compilation are detailed in Table 5.5. The depth of the QAOA-inspired circuit is markedly greater, with a depth of 44 as opposed to the 19 observed in the QPL circuit, indicating a surge of approximately 132%. The QAOA-inspired circuit also witnesses a substantial increment of about 71% compared to the QPL circuit in terms of the total gate count. This escalation in complexity is partially attributed to the inherent topology of the quantum hardware, where not all three qubits are mutually interconnected. Consequently, to compile the QAOA-inspired circuit into a hardware-compatible configuration, SWAP operations are implemented, which facilitates the establishment of effective communication between non-adjacent qubits. However, this measure contributes to the increased gate count, with the total gate count reaching 60, in contrast to the 35 gates required for the QPL circuit, marking a growth of approximately 71%. Especially the number

Table 5.5: Quantum Circuit Specifications after compilation

Name	Qubit number	Depth	Gate Counts				
			rz	sx	cx	measure	total
QPL	2	19	18	12	3	2	35
QAOA-inspired	3	44	26	14	18	2	60

of CNOT gates has increased sixfold. Considering that the error associated with two-qubit gates, namely CNOT gates, is significantly larger than that with single-qubit gates in quantum hardware, the increased number of CNOT gates dramatically amplifies the error impact on the circuit performance. This potentially renders the QAOA-inspired circuit impractical for implementation on quantum hardware. In contrast, the QPL circuit proposed in this study exhibits a high degree of consistency in performance between the quantum hardware and the simulator, as substantiated by the experimental results presented in Sec.5.5.2.

5.6 Summary

This study proposes a QPL algorithm to guide the online arbitrage of ESSs in the energy market. A carefully designed MDP encourages ESSs to actively explore the action space in the early stages of training while also ensuring the safe operation of the ESSs. A multi-layered VQC, characterized by low width and depth, is used as the policy, complemented with the derivation of the loss function and gradients to train the parameters of the circuit, namely, the rotation angles of the quantum rotation gates.

In comparison with the classic SAC algorithm, the proposed method demonstrates quantum superiority, attributable to the high expressivity of the designed VQC. With only 1.6% of the parameter quantity required by classical algorithms, QPL can achieve a faster convergence speed and optimization performance that is very close to that of classic algorithms. The trained VQCs are ported to run online on the IBM_Lagos quantum hardware, and the results indicate that gate errors and qubit decoherence exert a relatively subtle influence on the designed VQCs. Compared to a series of circuits proposed in other literature, this study validates the superiority of the proposed low-width and low-depth Quantum Policy circuits in terms of noise resistance, optimization performance, training stability, and compilation efficiency. The comparative results not only showcase the reliability of QPL but also signal a promising avenue for leveraging quantum computing capabilities to further integrate ESSs into the energy system.

Chapter 6
Concluding Remarks

In this dissertation, the emerging challenges in the integration of renewable energy sources and energy storage systems into power systems, driven by net-zero objectives, are addressed through several advanced methodologies. The research first introduces a Proximal Policy Optimization-based deep reinforcement learning agent tailored for the capacity scheduling of investor-owned photovoltaic-battery storage systems, ensuring adaptability to volatile market signals and photovoltaic generation profiles. Subsequently, an adaptive equivalent model for active distribution networks is proposed, which employs a leaves-trimming topological reduction method and a distributed Proximal Policy Optimization agent, enhancing the accuracy of models in scenarios with high renewable energy source penetration. Further, a bi-level multi-period framework is presented for strategic battery energy storage expansion planning in microgrids. This framework integrates quantile regression deep reinforcement learning with linear programming, aiming to provide dynamic planning solutions in the face of fluctuating renewable energy sources, load demands, and battery prices. Lastly, the dissertation proposes a quantum policy learning algorithm based on a strategically constructed variational quantum circuit, positioning it as a cutting-edge solution for energy arbitrage tasks in energy storage systems. Rigorous case studies, leveraging real electricity market data and renewable energy source profiles, validate the superiority of the proposed methodologies over contemporary techniques.

BIBLIOGRAPHY

[1] S. Nalley and A. LaRose, “Annual energy outlook 2022 (aeo2022),” *Energy Information Agency*, p. 23, 2022. 1

[2] G. He, J. Michalek, S. Kar, Q. Chen, D. Zhang, and J. F. Whitacre, “Utility-scale portable energy storage systems,” *Joule*, vol. 5, no. 2, pp. 379–392, 2021. 1, 71

[3] W. Cole, A. W. Frazier, and C. Augustine, “Cost projections for utility-scale battery storage: 2021 update,” National Renewable Energy Lab.(NREL), Golden, CO (United States), Tech. Rep., 2021. 1, 71, 86, 93, 97

[4] J.-H. Hong, D.-Y. Hong, L.-H. Yao, and L.-C. Fu, “A demand side management with appliance controllability analysis in smart home,” in *2020 International Conference on Smart Grids and Energy Systems (SGES)*. IEEE, 2020, pp. 556–561. 1

[5] L.-H. Yao, K.-C. Leung, C.-L. Tsai, C.-H. Huang, and L.-C. Fu, “A novel deep learning–based system for triage in the emergency department using electronic medical records: Retrospective cohort study,” *Journal of Medical Internet Research*, vol. 23, no. 12, p. e27008, 2021. 1

[6] R. Hunt, S. Kantra, D. Novosel, J. R. Aguero, D. W. Dietmeyer, and T. Rahman, “Roadmap for distribution synchronized measurements,” in *2022 International Conference on Smart Grid Synchronized Measurements and Analytics (SGSMA)*. IEEE, 2022, pp. 1–6. 2

[7] Y. Wang, Q. Chen, T. Hong, and C. Kang, “Review of smart meter data analytics: Applications, methodologies, and challenges,” *IEEE Transactions on Smart Grid*, vol. 10, no. 3, pp. 3125–3148, 2018. 2

[8] M. A. Hassan, A. Khalil, and M. Abubakr, “Selection methodology of representative meteorological days for assessment of renewable energy systems,” *Renewable Energy*, vol. 177, pp. 34–51, 2021. 2

[9] X. Huang, Z. Qin, and H. Liu, “A survey on power grid cyber security: From component-wise vulnerability assessment to system-wide impact analysis,” *IEEE Access*, vol. 6, pp. 69 023–69 035, 2018. 2

[10] Y. Yang, J. Xu, Z. Xu, P. Zhou, and T. Qiu, “Quantile context-aware social iot service big data recommendation with d2d communication,” *IEEE Internet of Things Journal*, vol. 7, no. 6, pp. 5533–5548, 2020. 2

[11] X. Huang, Z. Qin, M. Xie, H. Liu, and L. Meng, "Defense of massive false data injection attack via sparse attack points considering uncertain topological changes," *Journal of Modern Power Systems and Clean Energy*, vol. 10, no. 6, pp. 1588–1598, 2021. 2

[12] X. Ma, A. Karimpour, and Y.-J. Wu, "Eliminating the impacts of traffic volume variation on before and after studies: a causal inference approach," *Journal of Intelligent Transportation Systems*, pp. 1–15, 2023. 2

[13] K. Balakrishnan, Q. Cheng, J. Lee, D. Jeong, E. Kim, and J. Kim, "6-4: Deep learning for classification of repairable defects in display panels using multi-modal data," in *SID Symposium Digest of Technical Papers*, vol. 54, no. 1. Wiley Online Library, 2023, pp. 58–61. 2

[14] D. Zhang and F. Zhou, "Self-supervised image denoising for real-world images with context-aware transformer," *IEEE Access*, vol. 11, pp. 14 340–14 349, 2023. 2

[15] F. Zhou, Z. Fu, and D. Zhang, "High dynamic range imaging with context-aware transformer," in *2023 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2023, pp. 1–8. 2

[16] Q. Cheng, C. Zhang, and X. Shen, "Estimation of energy and time usage in 3d printing with multimodal neural network," in *2022 4th International Conference on Frontiers Technology of Information and Computer (ICFTIC)*. IEEE, 2022, pp. 900–903. 2

[17] M. Khodayar, G. Liu, J. Wang, and M. E. Khodayar, "Deep learning in power systems research: A review," *CSEE Journal of Power and Energy Systems*, vol. 7, no. 2, pp. 209–220, 2020. 2

[18] D. Zhang, F. Zhou, Y. Wei, X. Yang, and Y. Gu, "Unleashing the power of self-supervised image denoising: A comprehensive review," *arXiv preprint arXiv:2308.00247*, 2023. 2

[19] D. Zhang, F. Zhou, Y. Jiang, and Z. Fu, "Mm-bsn: Self-supervised image denoising for real-world with multi-mask based on blind-spot network," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 4188–4197. 2

[20] D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, L. Baker, M. Lai, A. Bolton *et al.*, "Mastering the game of go without human knowledge," *nature*, vol. 550, no. 7676, pp. 354–359, 2017. 2, 9

[21] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko *et al.*, "Highly accurate protein structure prediction with alphafold," *Nature*, vol. 596, no. 7873, pp. 583–589, 2021. 2

[22] A. Fawzi, M. Balog, A. Huang, T. Hubert, B. Romera-Paredes, M. Barekatain, A. Novikov, F. J. R Ruiz, J. Schrittwieser, G. Swirszcz *et al.*, "Discovering faster matrix multiplication algorithms with reinforcement learning," *Nature*, vol. 610, no. 7930, pp. 47–53, 2022. 2

[23] C. Zhang and Q. Cheng, "Predicting melt-crystal interface position and shape during the manufacturing process of single crystal via explainable machine learning models," in *IOP Conference Series: Materials Science and Engineering*, vol. 1258, no. 1. IOP Publishing, 2022, p. 012029. 3

[24] Q. Cheng, S. Qu, and J. Lee, "Shapnn: Shapley value regularized tabular neural network," *arXiv preprint arXiv:2309.08799*, 2023. 3

[25] J. Zhuang and M. Al Hasan, "Defending graph convolutional networks against dynamic graph perturbations via bayesian self-supervision," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, no. 4, 2022, pp. 4405–4413. 3

[26] ——, "Non-exhaustive learning using gaussian mixture generative adversarial networks," in *Machine Learning and Knowledge Discovery in Databases. Research Track: European Conference, ECML PKDD 2021, Bilbao, Spain, September 13–17, 2021, Proceedings, Part II 21*. Springer, 2021, pp. 3–18. 3

[27] B. Huang and J. Wang, "Applications of physics-informed neural networks in power systems-a review," *IEEE Transactions on Power Systems*, 2022. 3

[28] X. Huang, Z. Ding, and Z. Zhang, "A guided deep reinforcement learning method for distribution voltage regulation via battery systems," in *2021 IEEE Power & Energy Society Innovative Smart Grid Technologies Conference (ISGT)*. IEEE, 2021, pp. 1–5. 3

[29] F. Arute, K. Arya, R. Babbush, D. Bacon, J. C. Bardin, R. Barends, R. Biswas, S. Boixo, F. G. Brandao, D. A. Buell *et al.*, "Quantum supremacy using a programmable superconducting processor," *Nature*, vol. 574, no. 7779, pp. 505–510, 2019. 3

[30] T. Monz, D. Nigg, E. A. Martinez, M. F. Brandl, P. Schindler, R. Rines, S. X. Wang, I. L. Chuang, and R. Blatt, "Realization of a scalable shor algorithm," *Science*, vol. 351, no. 6277, pp. 1068–1070, 2016. 3

[31] A. Mandviwalla, K. Ohshiro, and B. Ji, "Implementing grover's algorithm on the ibm quantum computers," in *2018 IEEE international conference on big data (big data)*. IEEE, 2018, pp. 2531–2537. 3

[32] B. Huang and J. Wang, "Adaptive static equivalences for active distribution networks with massive renewable energy integration: A distributed deep reinforcement learning approach," *IEEE Transactions on Network Science and Engineering*, pp. 1–13, 2023. 3

[33] B. Huang, T. Zhao, M. Yue, and J. Wang, "Bi-level adaptive storage expansion strategy for microgrids using deep reinforcement learning," *IEEE Transactions on Smart Grid*, pp. 1–1, 2023. 3

[34] B. Huang and J. Wang, "Deep-reinforcement-learning-based capacity scheduling for pv-battery storage system," *IEEE Transactions on Smart Grid*, vol. 12, no. 3, pp. 2272–2283, 2021. 3, 107

[35] C. Byers and A. Botterud, "Additional capacity value from synergy of variable renewable energy and energy storage," *IEEE Transactions on Sustainable Energy*, 2019. 8

[36] "Cost projections for utility-scale battery storage," [EB/OL], https://www.nrel.gov/docs/fy19osti/73222.pdf Accessed May 2, 2020. 8

[37] F. Conte, S. Massucco, G. P. Schiapparelli, and F. Silvestro, "Day-ahead and intra-day planning of integrated bess-pv systems providing frequency regulation," *IEEE Transactions on Sustainable Energy*, 2019. 8, 9

[38] X. Huang, Z. Zhang, Y. Lin, and Y. Chen, "Arbitrage and capacity firming in coordination with day-ahead bidding of a hybrid pv plant," in *2022 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE, 2022, pp. 1–5. 8

[39] Y. Shi, B. Xu, D. Wang, and B. Zhang, "Using battery storage for peak shaving and frequency regulation: Joint optimization for superlinear gains," *IEEE Transactions on Power Systems*, vol. 33, no. 3, pp. 2882–2894, 2017. 8, 9, 20

[40] G. He, Q. Chen, C. Kang, P. Pinson, and Q. Xia, "Optimal bidding strategy of battery storage in power markets considering performance-based regulation and battery cycle life," *IEEE Transactions on Smart Grid*, vol. 7, no. 5, pp. 2359–2367, 2016. 8, 9, 15, 33

[41] J. Cao, D. Harrold, Z. Fan, T. Morstyn, D. Healey, and K. Li, "Deep reinforcement learning based energy storage arbitrage with accurate lithium-ion battery degradation model," *IEEE Transactions on Smart Grid*, 2020. 9, 10, 28, 29, 107

[42] Y. Wang, C. Wan, Z. Zhou, K. Zhang, and A. Botterud, "Improving deployment availability of energy storage with data-driven agc signal models," *IEEE Transactions on Power Systems*, vol. 33, no. 4, pp. 4207–4217, 2017. 9

[43] R. Kumar, M. J. Wenzel, M. J. Ellis, M. N. ElBsat, K. H. Drees, and V. M. Zavala, "A stochastic model predictive control framework for stationary battery systems," *IEEE Transactions on Power Systems*, vol. 33, no. 4, pp. 4397–4406, 2018. 9

[44] Y. Mei, T. Lan, M. Imani, and S. Subramaniam, "A bayesian optimization framework for finding local optima in expensive multi-modal functions," *arXiv preprint arXiv:2210.06635*, 2022. 9

[45] Y. Shi, B. Xu, Y. Tan, D. Kirschen, and B. Zhang, “Optimal battery control under cycle aging mechanisms in pay for performance settings,” *IEEE Transactions on Automatic Control*, vol. 64, no. 6, pp. 2324–2339, 2018. 9

[46] W. Wang, N. Yu, Y. Gao, and J. Shi, “Safe off-policy deep reinforcement learning algorithm for volt-var control in power distribution systems,” *IEEE Transactions on Smart Grid*, 2019. 9

[47] Y. Ye, D. Qiu, X. Wu, G. Strbac, and J. Ward, “Model-free real-time autonomous control for a residential multi-energy system using deep reinforcement learning,” *IEEE Transactions on Smart Grid*, 2020. 9

[48] H. Li, Z. Wan, and H. He, “Constrained ev charging scheduling based on safe deep reinforcement learning,” *IEEE Transactions on Smart Grid*, 2019. 9

[49] H. Wang and B. Zhang, “Energy storage arbitrage in real-time markets via reinforcement learning,” in *2018 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE, 2018, pp. 1–5. 10

[50] V.-H. Bui, A. Hussain, and H.-M. Kim, “Double deep q-learning-based distributed operation of battery energy storage system considering uncertainties,” *IEEE Transactions on Smart Grid*, vol. 11, no. 1, pp. 457–469, 2019. 10

[51] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” *arXiv preprint arXiv:1707.06347*, 2017. 10, 21, 24, 25, 27, 52, 53

[52] J. Schulman, S. Levine, P. Abbeel, M. Jordan, and P. Moritz, “Trust region policy optimization,” in *International conference on machine learning*, 2015, pp. 1889–1897. 10

[53] J. Schulman, P. Moritz, S. Levine, M. Jordan, and P. Abbeel, “High-dimensional continuous control using generalized advantage estimation,” *arXiv preprint arXiv:1506.02438*, 2015. 10, 24

[54] X. Luo, X. Ma, M. Munden, Y.-J. Wu, and Y. Jiang, “A multisource data approach for estimating vehicle queue length at metered on-ramps,” *Journal of Transportation Engineering, Part A: Systems*, vol. 148, no. 2, p. 04021117, 2022. 12

[55] X. Wang and J. Wang, “Economic assessment for battery swapping station-based frequency regulation service,” *IEEE Transactions on Industry Applications*, vol. PP, pp. 1–1, 04 2020. 15, 33

[56] X. Wang, J. Wang, and J. Liu, “V2g frequency regulation capacity optimal scheduling for battery swapping station using deep q-network,” *IEEE Transactions on Industrial Informatics*, 2020. 15

[57] X. Ma, A. Karimpour, and Y.-J. Wu, "Statistical evaluation of data requirement for ramp metering performance assessment," *Transportation Research Part A: Policy and Practice*, vol. 141, pp. 248–261, 2020. 23

[58] "Energy & ancillary services market operations," [EB/OL], https://www.pjm.com/library/manuals.aspx Accessed May 2, 2020. 27, 33

[59] "Solar power data for integration studies," [EB/OL], https://www.nrel.gov/grid/solar-power-data.html Accessed May 2, 2020. 28

[60] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra, "Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015. 29

[61] L. Engstrom, A. Ilyas, S. Santurkar, D. Tsipras, F. Janoos, L. Rudolph, and A. Madry, "Implementation matters in deep policy gradients: A case study on ppo and trpo," *arXiv preprint arXiv:2005.12729*, 2020. 30

[62] C. Szepesvári, "Algorithms for reinforcement learning," *Synthesis lectures on artificial intelligence and machine learning*, vol. 4, no. 1, pp. 1–103, 2010. 31

[63] J. N. Tsitsiklis and B. Van Roy, "An analysis of temporal-difference learning with function approximation," *IEEE Transactions on Automatic Control*, vol. 42, no. 5, pp. 674–690, 1997. 31

[64] G. He, J. Lin, F. Sifuentes, X. Liu, N. Abhyankar, and A. Phadke, "Rapid cost decrease of renewables and storage accelerates the decarbonization of china's power system," *Nature communications*, vol. 11, no. 1, pp. 1–9, 2020. 39

[65] Z. Zhang, Y. Zhang, D. Yue, C. Dou, L. Ding, and D. Tan, "Voltage regulation with high penetration of low-carbon energy in distribution networks: A source–grid–load-collaboration-based perspective," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 6, pp. 3987–3999, 2022. 39

[66] X. Shang, Z. Li, J. Zheng, and Q. Wu, "Equivalent modeling of active distribution network considering the spatial uncertainty of renewable energy resources," *International Journal of Electrical Power & Energy Systems*, vol. 112, pp. 83–91, 2019. 39, 41, 42

[67] T. Lv and Q. Ai, "Interactive energy management of networked microgrids-based active distribution system considering large-scale integration of renewable energy resources," *Applied Energy*, vol. 163, pp. 408–422, 2016. 40

[68] S. Rao and D. Tylavsky, "Nonlinear network reduction for distribution networks using the holomorphic embedding method," in *2016 North American Power Symposium (NAPS)*. IEEE, 2016, pp. 1–6. 40, 41

[69] W. Dai, J. Yu, X. Liu, and W. Li, "Two-tier static equivalent method of active distribution networks considering sensitivity, power loss and static load characteristics," *International Journal of Electrical Power & Energy Systems*, vol. 100, pp. 193–200, 2018. 40, 41

[70] Z. K. Pecenak, V. R. Disfani, M. J. Reno, and J. Kleissl, "Inversion reduction method for real and complex distribution feeder models," *IEEE Transactions on Power Systems*, vol. 34, no. 2, pp. 1161–1170, 2018. 40, 41, 50

[71] Z. K. Pecenak, H. V. Haghi, C. Li, M. J. Reno, V. R. Disfani, and J. Kleissl, "Aggregation of voltage-controlled devices during distribution network reduction," *IEEE Transactions on Smart Grid*, 2020. 40, 41, 50

[72] A. Samadi, L. Söder, E. Shayesteh, and R. Eriksson, "Static equivalent of distribution grids with high penetration of pv systems," *IEEE Transactions on Smart Grid*, vol. 6, no. 4, pp. 1763–1774, 2015. 40, 41, 61, 62, 67

[73] F. Mahmood, H. Hooshyar, J. Lavenius, A. Bidadfar, P. Lund, and L. Vanfretti, "Real-time reduced steady-state model synthesis of active distribution networks using pmu measurements," *IEEE Transactions on Power Delivery*, vol. 32, no. 1, pp. 546–555, 2017. 40, 41

[74] B. Huang, Z. Li, J. Zheng, and Q. Wu, "Probabilistic active distribution network equivalence with correlated uncertain injections for grid analysis," *IET Renewable Power Generation*, vol. 14, no. 11, pp. 1964–1977, 2020. 41

[75] B. Huang, P. Li, J. Zheng, and Q. Wu, "A modified ward equivalent based on sensitivity matrices for static security analysis," *IEEJ Transactions on Electrical and Electronic Engineering*, vol. 13, no. 11, pp. 1675–1676, 2018. 41

[76] Y. Ji, X. Zhang, X. Wang, X. Huang, B. Huang, J. H. Zheng, and Z. Li, "An equivalent modeling method for multi-port area load based on the extended generalized zip load model," in *2018 International Conference on Power System Technology (POWERCON)*, 2018, pp. 553–558. 41

[77] B. Huang, X. Shang, J. Zheng, Z. Li, Q. Wu, and X. Zhou, "Electrical network equivalent modeling method with boundary buses interconnected," in *2019 IEEE PES GTD Grand International Conference and Exposition Asia (GTD Asia)*. IEEE, 2019, pp. 429–434. 41

[78] G. Chaspierre, G. Denis, P. Panciatici, and T. Van Cutsem, "An active distribution network equivalent derived from large-disturbance simulations with uncertainty," *IEEE Transactions on Smart Grid*, vol. 11, no. 6, pp. 4749–4759, 2020. 41

[79] G. Mitrentsis and H. Lens, "Data-driven dynamic models of active distribution networks using unsupervised learning techniques on field measurements," *IEEE Transactions on Smart Grid*, vol. 12, no. 4, pp. 2952–2965, 2021. 41

[80] B. Huang and J. Wang, "Deep-reinforcement-learning-based capacity scheduling for pv-battery storage system," *IEEE Transactions on Smart Grid*, vol. 12, no. 3, pp. 2272–2283, 2021. 41

[81] Y. Zhao, Q. Pan, K. Choromanski, D. Jain, and V. Sindhwani, "Implicit two-tower policies," *arXiv preprint arXiv:2208.01191*, 2022. 41

[82] Y. Wang, B. Chai, W. Lu, and X. Zheng, "A review of deep reinforcement learning applications in power system parameter estimation," in *2021 International Conference on Power System Technology (POWERCON)*. IEEE, 2021, pp. 2015–2021. 41

[83] X. Wang, Y. Wang, D. Shi, J. Wang, and Z. Wang, "Two-stage wecc composite load modeling: A double deep q-learning networks approach," *IEEE Transactions on Smart Grid*, vol. 11, no. 5, pp. 4331–4344, 2020. 41

[84] S. Wang, R. Diao, C. Xu, D. Shi, and Z. Wang, "On multi-event co-calibration of dynamic model parameters using soft actor-critic," *IEEE Transactions on Power Systems*, vol. 36, no. 1, pp. 521–524, 2021. 42

[85] G. Zhang, W. Hu, J. Zhao, D. Cao, Z. Chen, and F. Blaabjerg, "A novel deep reinforcement learning enabled multi-band pss for multi-mode oscillation control," *IEEE Transactions on Power Systems*, vol. 36, no. 4, pp. 3794–3797, 2021. 42

[86] C. Jiang, Z. Li, J. Zheng, Q. Wu, and X. Shang, "Two-level area-load modelling for opf of power system using reinforcement learning," *IET Generation, Transmission & Distribution*, vol. 13, no. 18, pp. 4141–4149, 2019. 42

[87] V. Mnih, A. P. Badia, M. Mirza, A. Graves, T. Lillicrap, T. Harley, D. Silver, and K. Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International conference on machine learning*. PMLR, 2016, pp. 1928–1937. 42

[88] "Ieee standard for interconnection and interoperability of distributed energy resources with associated electric power systems interfaces," *IEEE Std 1547-2018 (Revision of IEEE Std 1547-2003)*, pp. 1–138, 2018. 45

[89] J. Zhang, Y. Wang, Y. Weng, and N. Zhang, "Topology identification and line parameter estimation for non-pmu distribution network: A numerical method," *IEEE Transactions on Smart Grid*, vol. 11, no. 5, pp. 4440–4453, 2021. 47

[90] X. Huang, J.-y. Gwak, H. Cui, L. Yu, and Z. Zhang, "Transient stability preventive control via tuning the parameters of virtual synchronous generators," in *2023 IEEE PES General Meeting (PESGM)*. IEEE, 2023, p. accepted. 48

[91] X. Ji, Z. Yin, Y. Zhang, M. Wang, X. Zhang, C. Zhang, and D. Wang, "Real-time robust forecasting-aided state estimation of power system based on data-driven models," *International Journal of Electrical Power & Energy Systems*, vol. 125, p. 106412, 2021. 50

[92] S. Meinecke, D. Sarajlić, S. R. Drauz, A. Klettke, L.-P. Lauven, C. Rehtanz, A. Moser, and M. Braun, "Simbench—a benchmark dataset of electric power systems to compare innovative solutions based on power flow analysis," *Energies*, vol. 13, no. 12, p. 3290, Jun. 2020. 57

[93] R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "Matpower: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on power systems*, vol. 26, no. 1, pp. 12–19, 2010. 65

[94] D. Cao, J. Zhao, W. Hu, F. Ding, Q. Huang, and Z. Chen, "Attention enabled multi-agent drl for decentralized volt-var control of active distribution system using pv inverters and svcs," *IEEE Transactions on Sustainable Energy*, vol. 12, no. 3, pp. 1582–1592, 2021. 70

[95] Y. Mei, H. Zhou, T. Lan, G. Venkataramani, and P. Wei, "Mac-po: Multi-agent experience replay via collective priority optimization," in *Proceedings of the 2023 International Conference on Autonomous Agents and Multiagent Systems*, 2023, pp. 466–475. 70

[96] H. Zhou, T. Lan, and V. Aggarwal, "Value functions factorization with latent state information sharing in decentralized multi-agent policy gradients," *IEEE Transactions on Emerging Topics in Computational Intelligence*, 2023. 70

[97] ——, "Pac: Assisted value factorization with counterfactual predictions in multi-agent reinforcement learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 15 757–15 769, 2022. 70

[98] J. M. Home-Ortiz, M. Pourakbari-Kasmaei, M. Lehtonen, and J. R. S. Mantovani, "A mixed integer conic model for distribution expansion planning: Matheuristic approach," *IEEE Transactions on Smart Grid*, vol. 11, no. 5, pp. 3932–3943, 2020. 72, 78

[99] J. Zheng, W. Xiao, C. Wu, Z. Li, L. Wang, and Q. Wu, "A gradient descent direction based-cumulants method for probabilistic energy flow analysis of individual-based integrated energy systems," *Energy*, vol. 265, p. 126290, 2023. 72

[100] S. He, Z. Zhang, S. Han, L. Pepin, G. Wang, D. Zhang, J. A. Stankovic, and F. Miao, "Data-driven distributionally robust electric vehicle balancing for autonomous mobility-on-demand systems under demand and supply uncertainties," *IEEE Transactions on Intelligent Transportation Systems*, 2023. 72

[101] Y. Yang, L. Chen, P. Zhou, and X. Ding, "Vflh: A following-the-leader-history based algorithm for adaptive online convex optimization with stochastic constraints," *Available at SSRN 4040704*, 2023. 72

[102] A. H. Alobaidi, M. Khodayar, A. Vafamehr, H. Gangammanavar, and M. E. Khodayar, "Stochastic expansion planning of battery energy storage for the interconnected distribution and data networks," *International Journal of Electrical Power & Energy Systems*, vol. 133, p. 107231, 2021. 72, 86, 96

[103] M. Asensio, P. M. de Quevedo, G. Muñoz-Delgado, and J. Contreras, "Joint distribution network and renewable energy expansion planning considering demand response and energy storage—part i: Stochastic programming model," *IEEE Transactions on Smart Grid*, vol. 9, no. 2, pp. 655–666, 2018. 72

[104] M. Nazemi, M. Moeini-Aghtaie, M. Fotuhi-Firuzabad, and P. Dehghanian, "Energy storage planning for enhanced resilience of power distribution networks against earthquakes," *IEEE Transactions on Sustainable Energy*, vol. 11, no. 2, pp. 795–806, 2020. 72

[105] X. Cao, T. Cao, F. Gao, and X. Guan, "Risk-averse storage planning for improving res hosting capacity under uncertain siting choices," *IEEE transactions on sustainable energy*, vol. 12, no. 4, pp. 1984–1995, 2021. 72

[106] A. M. Nakiganda, S. Dehghan, U. Markovic, G. Hug, and P. Aristidou, "A stochastic-robust approach for resilient microgrid investment planning under static and transient islanding security constraints," *IEEE Transactions on Smart Grid*, vol. 13, no. 3, pp. 1774–1788, 2022. 72

[107] I. Alsaidan, A. Khodaei, and W. Gao, "A comprehensive battery energy storage optimal sizing model for microgrid applications," *IEEE Transactions on Power Systems*, vol. 33, no. 4, pp. 3968–3980, 2018. 72

[108] R. Xie, W. Wei, M. Li, Z. Dong, and S. Mei, "Sizing capacities of renewable generation, transmission, and energy storage for low-carbon power systems: A distributionally robust optimization approach," *Energy*, vol. 263, p. 125653, 2023. 72

[109] Y. Bengio, A. Lodi, and A. Prouvost, "Machine learning for combinatorial optimization: a methodological tour d'horizon," *European Journal of Operational Research*, vol. 290, no. 2, pp. 405–421, 2021. 72

[110] B. Huang and J. Wang, "Deep-reinforcement-learning-based capacity scheduling for pv-battery storage system," *IEEE Transactions on Smart Grid*, vol. 12, no. 3, pp. 2272–2283, 2021. 72

[111] S. He, Y. Wang, S. Han, S. Zou, and F. Miao, "A robust and constrained multi-agent reinforcement learning framework for electric vehicle amod systems," *arXiv preprint arXiv:2209.08230*, 2022. 72

[112] S. Tsianikas, N. Yousefi, J. Zhou, M. D. Rodgers, and D. Coit, "A storage expansion planning framework using reinforcement learning and simulation-based optimization," *Applied Energy*, vol. 290, p. 116778, 2021. 73, 78

[113] K. Pang, J. Zhou, S. Tsianikas, and Y. Ma, "Deep reinforcement learning for resilient microgrid expansion planning with multiple energy resource," *Quality and Reliability Engineering International*, 2022. 73

[114] X. Ma, "Traffic performance evaluation using statistical and machine learning methods," Ph.D. dissertation, The University of Arizona, 2022. 73

[115] J. E. Contreras-Ocaña, A. Singh, Y. Bésanger, and F. Wurtz, "Integrated planning of a solar/storage collective," *IEEE Transactions on Smart Grid*, vol. 12, no. 1, pp. 215–226, 2021. 78

[116] H. Karimianfard, H. Haghighat, and B. Zeng, "Co-optimization of battery storage investment and grid expansion in integrated energy systems," *IEEE Systems Journal*, vol. 16, no. 4, pp. 5928–5938, 2022. 78

[117] G. Cao, Z. Lu, X. Wen, T. Lei, and Z. Hu, "Aif: An artificial intelligence framework for smart wireless network management," *IEEE Communications Letters*, vol. 22, no. 2, pp. 400–403, 2017. 79

[118] S. H. Low, "Convex relaxation of optimal power flow—part i: Formulations and equivalence," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 1, pp. 15–27, 2014. 80

[119] O. Ciftci, M. Mehrtash, and A. Kargarian, "Data-driven nonparametric chance-constrained optimization for microgrid energy management," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 4, pp. 2447–2457, 2020. 82

[120] Y. Wen, C. Chung, X. Liu, and L. Che, "Microgrid dispatch with frequency-aware islanding constraints," *IEEE Transactions on Power Systems*, vol. 34, no. 3, pp. 2465–2468, 2019. 82, 83

[121] H. Chávez, R. Baldick, and S. Sharma, "Governor rate-constrained opf for primary frequency control adequacy," *IEEE Transactions on Power Systems*, vol. 29, no. 3, pp. 1473–1480, 2014. 82, 84

[122] Y. Wen, W. Li, G. Huang, and X. Liu, "Frequency dynamics constrained unit commitment with battery energy storage," *IEEE Transactions on Power Systems*, vol. 31, no. 6, pp. 5115–5125, 2016. 82, 83, 84

[123] M. Hessel, J. Modayil, H. Van Hasselt, T. Schaul, G. Ostrovski, W. Dabney, D. Horgan, B. Piot, M. Azar, and D. Silver, "Rainbow: Combining improvements in deep reinforcement learning," in *Thirty-second AAAI conference on artificial intelligence*, 2018. 89, 90

[124] W. Dabney, M. Rowland, M. Bellemare, and R. Munos, "Distributional reinforcement learning with quantile regression," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018. 90

[125] B. Huang, "renewable load profile and network parameter," https://figshare.com/s/f1e68d3d34321247b261, accessed: 2023-04-20. 93

[126] C. Li, A. J. Conejo, J. D. Siirola, and I. E. Grossmann, "On representative day selection for capacity expansion planning of power systems under extreme operating conditions," *International Journal of Electrical Power & Energy Systems*, vol. 137, p. 107697, 2022. 94

[127] X. Ma, A. Karimpour, and Y.-J. Wu, “On-ramp and off-ramp traffic flows estimation based on a data-driven transfer learning framework,” *arXiv preprint arXiv:2308.03538*, 2023. 105

[128] X. Ma, A. Cottam, M. R. R. Shaon, and Y.-J. Wu, “A transfer learning framework for proactive ramp metering performance assessment,” *arXiv preprint arXiv:2308.03542*, 2023. 105

[129] T. Zhao, A. Parisio, and J. V. Milanović, “Distributed control of battery energy storage systems in distribution networks for voltage regulation at transmission–distribution network interconnection points,” *Control Engineering Practice*, vol. 119, p. 104988, 2022. 106

[130] L. Feng, X. Zhang, C. Li, X. Li, B. Li, J. Ding, C. Zhang, H. Qiu, Y. Xu, and H. Chen, “Optimization analysis of energy storage application based on electricity price arbitrage and ancillary services,” *Journal of Energy Storage*, vol. 55, p. 105508, 2022. 107

[131] Y. Yang, C. Liu, and Z. Zhang, “Particle-based online bayesian sampling,” *arXiv preprint arXiv:2302.14796*, 2023. 107

[132] A. N. Elmachtoub, H. Lam, H. Zhang, and Y. Zhao, “Estimate-then-optimize versus integrated-estimation-optimization: A stochastic dominance perspective,” *arXiv preprint arXiv:2304.06833*, 2023. 107

[133] H. Su, D. Feng, Y. Zhao, Y. Zhou, Q. Zhou, C. Fang, and U. Rahman, “Optimization of customer-side battery storage for multiple service provision: Arbitrage, peak shaving, and regulation,” *IEEE Transactions on Industry Applications*, vol. 58, no. 2, pp. 2559–2573, 2022. 107

[134] M. U. Hashmi, D. Deka, A. Bušić, L. Pereira, and S. Backhaus, “Arbitrage with power factor correction using energy storage,” *IEEE Transactions on Power Systems*, vol. 35, no. 4, pp. 2693–2703, 2020. 107

[135] S. Hanif, M. J. E. Alam, K. Roshan, B. A. Bhatti, and J. C. Bedoya, “Multi-service battery energy storage system optimization and control,” *Applied Energy*, vol. 311, p. 118614, 2022. 107

[136] D. J. Harrold, J. Cao, and Z. Fan, “Data-driven battery operation for energy arbitrage using rainbow deep reinforcement learning,” *Energy*, vol. 238, p. 121958, 2022. 107

[137] A. Elmachtoub, V. Gupta, and Y. Zhao, “Balanced off-policy evaluation for personalized pricing,” in *Proceedings of The 26th International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, F. Ruiz, J. Dy, and J.-W. van de Meent, Eds., vol. 206. PMLR, 25–27 Apr 2023, pp. 10 901–10 917. [Online]. Available: https://proceedings.mlr.press/v206/elmachtoub23a.html 108

[138] A. Abbas, D. Sutter, C. Zoufal, A. Lucchi, A. Figalli, and S. Woerner, “The power of quantum neural networks,” *Nature Computational Science*, vol. 1, no. 6, pp. 403–409, 2021. 108

[139] M. Y. Niu, A. Zlokapa, M. Broughton, S. Boixo, M. Mohseni, V. Smelyanskyi, and H. Neven, “Entangling quantum generative adversarial networks,” *Physical Review Letters*, vol. 128, no. 22, p. 220505, 2022. 108

[140] Y. Zhou and P. Zhang, “Noise-resilient quantum machine learning for stability assessment of power systems,” *IEEE Transactions on Power Systems*, vol. 38, no. 1, pp. 475–487, 2023. 108

[141] Y. Du, M.-H. Hsieh, T. Liu, and D. Tao, “Expressive power of parametrized quantum circuits,” *Physical Review Research*, vol. 2, no. 3, p. 033125, 2022. 108, 129

[142] S. Y.-C. Chen, C.-H. H. Yang, J. Qi, P.-Y. Chen, X. Ma, and H.-S. Goan, “Variational quantum circuits for deep reinforcement learning,” *IEEE Access*, vol. 8, pp. 141 007–141 024, 2020. 108

[143] Q. Lan, “Variational quantum soft actor-critic,” *arXiv preprint arXiv:2112.11921*, 2021. 108, 121

[144] N. Meyer, D. Scherer, A. Plinge, C. Mutschler, and M. Hartmann, “Quantum policy gradient algorithm with optimized action decoding,” in *International Conference on Machine Learning*. PMLR, 2023, pp. 24 592–24 613. 108, 109

[145] R. Yan, Y. Wang, Y. Xu, and J. Dai, “A multiagent quantum deep reinforcement learning method for distributed frequency control of islanded microgrids,” *IEEE Transactions on Control of Network Systems*, vol. 9, no. 4, pp. 1622–1632, 2022. 109

[146] W. J. Yun, J. P. Kim, S. Jung, J.-H. Kim, and J. Kim, “Quantum multi-agent actor-critic neural networks for internet-connected multi-robot coordination in smart factory management,” *IEEE Internet of Things Journal*, 2023. 109

[147] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor,” in *International conference on machine learning*. PMLR, 2018, pp. 1861–1870. 115

[148] M. Jankowiak and F. Obermeyer, “Pathwise derivatives beyond the reparameterization trick,” in *International conference on machine learning*. PMLR, 2018, pp. 2235–2244. 126

[149] L. Funcke, T. Hartung, K. Jansen, S. Kühn, and P. Stornati, “Dimensional expressivity analysis of parametric quantum circuits,” *Quantum*, vol. 5, p. 422, 2021. 129

[150] S. Lloyd, M. Schuld, A. Ijaz, J. Izaac, and N. Killoran, “Quantum embeddings for machine learning,” *arXiv preprint arXiv:2001.03622*, 2020. 135, 137

[151] V. Havlíček, A. D. Córcoles, K. Temme, A. W. Harrow, A. Kandala, J. M. Chow, and J. M. Gambetta, "Supervised learning with quantum-enhanced feature spaces," *Nature*, vol. 567, no. 7747, pp. 209–212, 2019. 135, 137

[152] M. J. Bremner, A. Montanaro, and D. J. Shepherd, "Achieving quantum supremacy with sparse and noisy commuting quantum computations," *Quantum*, vol. 1, p. 8, 2017. 137