TU Delft

Delft University of Technology

Deep Reinforcement Learning Based Optimal Distribution Networks Operation

Shengren, H.

DOI
10.4233/uuid:e47b3b57-b183-4ef7-a5cc-30773cd8c957

Publication date
2025

Document Version
Final published version

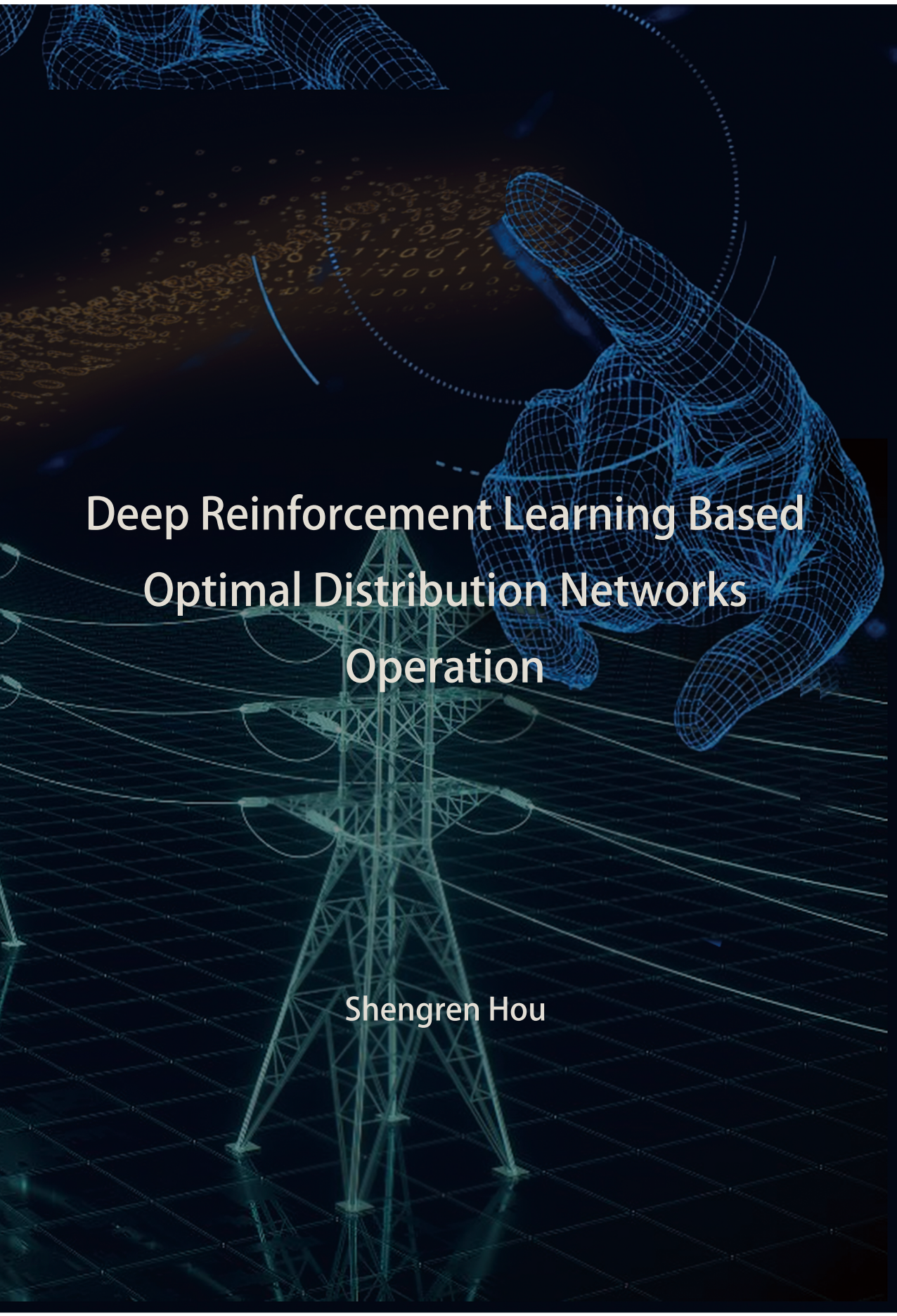
Citation (APA)
Shengren, H. (2025). *Deep Reinforcement Learning Based Optimal Distribution Networks Operation*. [Dissertation (TU Delft), Delft University of Technology]. https://doi.org/10.4233/uuid:e47b3b57-b183-4ef7-a5cc-30773cd8c957

Important note
To cite this publication, please use the final published version (if applicable).
Please check the document version above.

Copyright
Other than for strictly personal use, it is not permitted to download, forward or distribute the text or part of it, without the consent of the author(s) and/or copyright holder(s), unless the work is under an open content license such as Creative Commons.

Takedown policy
Please contact us and provide details if you believe this document breaches copyrights.
We will remove access to the work immediately and investigate your claim.

This work is downloaded from Delft University of Technology.
For technical reasons the number of authors shown on this cover page is limited to a maximum of 10.



Deep Reinforcement Learning Based Optimal Distribution Networks Operation

Shengren Hou

Deep RL for Optimal Distribution Networks Operation

PhD Defence– October 27, 2025, Aula, TU Delft

Shengren Hou

Deep Reinforcement Learning Based Optimal Distribution Networks Operation

DEEP REINFORCEMENT LEARNING BASED OPTIMAL DISTRIBUTION NETWORKS OPERATION

Dissertation

for the purpose of obtaining the degree of doctor
at Delft University of Technology,
by the authority of the Rector Magnificus Prof.dr.ir. T.H.J.J. van der Hagen,
Chair of the Board for Doctorates
to be defended publicly on
Monday 27 October 2025 at 15:00 o'clock

by

Shengren HOU

Master of Science in Electrical Engineering, Guangxi University, China
born in Henan, China.

The dissertation has been approved by the promotors.

Promotor: Prof. dr. P. Palensky
Copromotor: Dr. P.P. Vergara Barrios

Composition of the doctoral committee:

Rector Magnificus,	chairperson
Prof. dr. P. Palensky,	Delft University of Technology, promotor
Dr. P.P. Vergara Barrios,	Delft University of Technology, copromotor

Independent members:

Prof. dr. M.T.J. Spaan,	Delft University of Technology
Prof. dr. D. Ernst,	University of Liège
Dr. P.H. Nguyen,	Eindhoven University of Technology
Prof. dr. A. Plaat,	University of Leiden
Prof. dr. M. Popov,	Delft University of Technology, Reserve member

This research is financially supported by China Scholarship Council (CSC).

Keywords: voltage control, reinforcement learning, distribution networks, energy management, sequential-decision-making

Printed by: Proefschrift Specialist

Cover design by: Shengren Hou

Copyright © 2024 by Shengren Hou

An electronic version of this dissertation is available at
http://repository.tudelft.nl/.

Dedicate to a better clean energy world.

May my humble efforts be a small part of the change.

Shengren HOU

NOTATION

List of Abbreviations

ADN	Active Distribution Network
ANM	Active Network Management
A3C	Asynchronous Advantage Actor-Critic
BESS	Battery Energy Storage System
BC	Behavior Cloning
CPO	Constrained Policy Optimization
DDPG	Deep Deterministic Policy Gradient
DERs	Distributed Energy Resources
DNN	Deep Neural Network
DN	Distribution Network
DRL	Deep Reinforcement Learning
DSO	Distribution System Operator
DPG	Deterministic Policy Gradient
EMS	Energy Management System
ESS	Energy Storage System
EV	Electric Vehicle
GAN	Generative Adversarial Network
GMC	Gaussian Mixture Models-Copula
IL	Imitation Learning
IRL	Imitation Reinforcement Learning
LP	Linear Programming
MDP	Markov Decision Process
MILP	Mixed Integer Linear Programming
MINL	Mixed Integer Nonlinear
MINLP	Mixed Integer Nonlinear Programming
MIP	Mixed Integer Programming
MIP-DQN	Mixed Integer Programming-Deep Q-Network
MIP-DRL	Mixed Integer Programming-Deep Reinforcement Learning
NLP	Non-linear Programming
OESS	Optimal energy system scheduling
PG	Policy Gradient
PPO	Proximal Policy Optimization
PV	Photovoltaic
ReLU	Rectified Linear Unit
RL	Reinforcement Learning
SAC	Soft Actor Critic

SOC	State of Charge
TD3	Twin Delayed Deep Deterministic Policy Gradient
ToU	Time-of-Use
TPF	Tensor Power Flow

CONTENTS

Notation vii

Summary xiii

Samenvatting xv

1 Introduction 1
- 1.1 Background and Motivation . . . 2
  - 1.1.1 Traditional Approaches and Limitations . . . 3
  - 1.1.2 The Emergence of Reinforcement Learning Approaches . . . 3
  - 1.1.3 Challenges . . . 5
- 1.2 Research Objective and Questions . . . 5
  - 1.2.1 Enforcing Distribution Networks Operational Constraints using DRL 5
  - 1.2.2 Leveraging Domain Knowledge to Ensure Safety, Increase Performance, and Enhance Computational Efficiency in DRL for Optimal DNs Operation . . . 6
  - 1.2.3 Reducing Computational Cost and Accelerating Training for DRL based Optimal DNs Operational Problems . . . 7
- 1.3 Contributions and Thesis Outline . . . 8

2 Optimal Energy System Scheduling Using A Constraint-Aware Reinforcement Learning Algorithm 11
- 2.1 Introduction . . . 12
  - 2.1.1 Literature Review . . . 12
  - 2.1.2 Contributions . . . 14
- 2.2 Mathematical Programming Formulation of the Energy Systems Scheduling Problem . . . 15
- 2.3 MDP Formulation & Value-Based DRL . . . 18
  - 2.3.1 DRL Value-Based Algorithms . . . 18
- 2.4 Proposed MIP-DQN Algorithm . . . 19
  - 2.4.1 Training Procedure . . . 20
  - 2.4.2 Deployment (Online Execution) Procedure . . . 20
- 2.5 Simulation Results and Discussions . . . 22
  - 2.5.1 Case Study and Simulations Setup . . . 23
  - 2.5.2 Validation and Algorithms for Comparison . . . 24
  - 2.5.3 Performance on the Training Set . . . 25
  - 2.5.4 Performance on the Test Set . . . 26
  - 2.5.5 Dispatch Decisions Comparison . . . 27
  - 2.5.6 Sensitivity Analysis . . . 28

2.5.7 Comparison with Safe DDPG Algorithm . . . 28
2.5.8 Larger Case Study . . . 31
2.5.9 Discussion . . . 33

3 MIP-DRL: A Constraint Enforcement Deep Reinforcement Learning Framework for Optimal Energy Storage System Dispatch 35
3.1 Introduction . . . 37
3.2 Mathematical Formulation . . . 39
3.3 ESSs Scheduling Problem MDP Formulation . . . 41
3.3.1 Operational Constraints . . . 42
3.4 Constraint Enforcement MIP-DRL Framework . . . 42
3.4.1 Step-by-Step Training . . . 42
3.4.2 Enforcing Constraints in Online Execution . . . 44
3.5 Simulation Results and Discussions . . . 46
3.5.1 Simulation Setup . . . 46
3.5.2 Performance of MIP-DRL algorithms on the Training Set . . . 47
3.5.3 Constraint Enforcement Capabilities and Performance . . . 48
3.5.4 Performance Comparison with Benchmarks . . . 48
3.5.5 Error Assessment and Computational Performance . . . 53
3.5.6 Scalability Analysis . . . 53
3.6 Discussion . . . 54

4 DistFlow Safe Reinforcement Learning Algorithm for Voltage Magnitude Regulation in Distribution Networks 55
4.1 Introduction . . . 56
4.2 Voltage Magnitude Regulation Problem . . . 58
4.2.1 Mathematical Programming Formulation . . . 58
4.2.2 CMDP Formulation . . . 60
4.3 Proposed DistFlow Safe RL Algorithm . . . 61
4.3.1 Deep Deterministic RL Algorithms . . . 61
4.3.2 Linear Power Flow Formulation . . . 62
4.3.3 Safety Layer Formulation . . . 63
4.3.4 Proposed DF-SRL algorithm . . . 63
4.4 Simulation Results and Discussions . . . 64
4.4.1 Simulations Setup, Data and Implementation . . . 64
4.4.2 Performance on the Training Set . . . 66
4.4.3 Performance and Constraint Enforcement Capabilities on Testing Set . . . 68
4.4.4 Sensitivity Analysis . . . 70
4.5 Scalability analysis . . . 71

5 Safe Imitation Learning-based Optimal Energy Storage Systems Dispatch in Distribution Networks 73
5.1 Introduction . . . 74
5.2 Mathematical Formulation . . . 76
5.3 MDP Formulation . . . 77

5.4 The Proposed Framework . . . 78
5.4.1 Offline Training Via Imitation Learning . . . 79
5.4.2 Online Execution with Safe Layer . . . 81
5.5 Simulation Results . . . 82
5.5.1 Performance on Training Set . . . 82
5.5.2 Dispatch Decision Comparision on Testing Dataset . . . 84
5.5.3 Scalability Analysis . . . 87
5.6 Discussion . . . 88

6 RL-ADN: A High-Performance Deep Reinforcement Learning Environment for Optimal Energy Storage Systems Dispatch 91
6.1 Introduction . . . 93
6.1.1 Motivation . . . 93
6.1.2 Related Work . . . 93
6.1.3 Contributions . . . 94
6.2 Background . . . 96
6.2.1 Optimal ESS dispatch tasks in distribution networks . . . 96
6.2.2 MDP formulation and reinforcement learning . . . 96
6.3 RL-ADN Framework . . . 98
6.3.1 Overview . . . 98
6.3.2 Data Source Layer . . . 98
6.3.3 Configuration Layer . . . 99
6.3.4 Interaction Loop Layer . . . 100
6.3.5 MDP Design . . . 101
6.3.6 Data Augmentation Model . . . 102
6.3.7 Laurent Power Flow . . . 102
6.4 Benchmark Scheme and Experiments . . . 103
6.4.1 Optimal ESSs dispatch Task and MDPs . . . 103
6.4.2 Bench-marking Approach . . . 104
6.5 Results . . . 104
6.5.1 Performance of DRL Algorithms on Template Optimal Dispatch Task 104
6.5.2 Impacts of Data Augmentation on Performance of DRL algorithms . 108
6.5.3 Enhancement of computation efficiency . . . 111
6.6 Discussion . . . 113

7 Conclusion and Discussion 115
7.1 Research Contributions to Research Questions . . . 116
7.1.1 Enforcing Distribution Network Operational Constraints using DRL (Q1): . . . 116
7.1.2 Leveraging Domain Knowledge for Safety, Performance, and Computational Efficiency (Q2): . . . 116
7.1.3 Reducing Computational Cost and Accelerating Training in DRL Algorithms (Q3): . . . 117
7.2 Discussion and Research Recommendations . . . 118

Bibliography **119**

Curriculum Vitæ **133**

List of Publications and Projects Involved **135**

Acknowledgements **137**

Appendix **141**
A Proof of MIP-DQN 141
B Workflows for Modules in RL-ADN 141
B.1 Data Manager Workflow 141
B.2 Data Augmentation Workflow 142

SUMMARY

The integration of distributed energy resources (DERs) and the increasing penetration of renewable energy generation have significantly increased the complexity and uncertainty of modern distribution networks. These developments necessitate advanced dispatch algorithms capable of handling the variability and operational constraints inherent in such systems. This thesis focuses on developing model-free deep reinforcement learning (DRL) algorithms to ensure reliable, safe, cost-effective operation in distribution networks (DNs). The research questions addressed in this thesis explore various challenges associated with the enforcement of operational constraints, learning efficiency, and computational cost reduction in DRL-based optimal operation of DNs.

First, the enforcement of power balance constraints is critical for maintaining system stability and reliability. Standard model-free DRL approaches often struggle with setting the strictly enforcement for system constraints. To address this, a DRL algorithm, MIP-DQN, was developed to strictly enforce all operational constraints in the action space, ensuring feasible dispatch in real-time operation. By leveraging optimization advances for deep neural networks (DNNs) that allow their representation as mixed-integer programming (MIP) formulations, this algorithm ensures that constraints are met even during online execution. Comparative performance evaluations with state-of-the-art DRL algorithms demonstrated the effectiveness of MIP-DQN in maintaining the power balance constraint.

Second, addressing sequential constraints in DRL-based optimal DNs operation requires dynamic adaptation to the timing and order of actions to ensure safe and reliable operations. Building on previous work, a Mixed-Integer Programming Deep Reinforcement Learning (MIP-DRL) framework was developed. This versatile framework enables various standard actor-critic DRL algorithms to enforce operational constraints strictly. By utilizing the robust constraint-enforcing ability of MIP, the MIP-DRL framework guarantees zero-constraint violations during online execution, extending the feasibility of real-time applications of DRL.

Third, the integration of operational constraints with prior knowledge into DRL frameworks improves solution feasibility and reduces problem complexity. This thesis introduced the DistFlow Safe Reinforcement Learning (DF-SRL) algorithm, which incorporates expert knowledge to accurately map the relationship between agent actions and voltage magnitude variations in DNs. The DF-SRL algorithm overlays a safety layer on top of the DRL policy to recalibrate potentially unsafe actions, ensuring strict enforcement of voltage magnitude constraints during both training and application phases.

Fourth, ensuring a safe and efficient algorithm to learn is crucial for practical implementation. A Safe Imitation Reinforcement Learning Framework combining Twin Delayed Deep Deterministic Policy Gradient (TD3) and imitation learning (IL) is proposed. This framework enhances performance and training efficiency while rigorously enforcing operational constraints. The offline training phase employs a dual-gradient strategy

using both behavior cloning (BC) policy and the critic network to stabilize training and expedite learning. A safe layer scrutinizes actions recommended by the trained TD3BC algorithm, filtering out unsafe actions and ensuring operational feasibility in scenarios not covered by expert data.

Finally, reducing computational costs and accelerating the training process are essential for the practical deployment of DRL algorithms in DNs operation. The RL-ADN library was introduced as an open-source tool specifically tailored for DRL-based optimal ESSs operation in distribution networks. RL-ADN provides extensive customization capabilities, integrating a novel data augmentation module using Gaussian Mixture Models-Copula (GMC) and the Tensor Power Flow (TPF) solver, significantly reducing computation time for power flow calculations. This library sets a new standard in DRL-based ESSs dispatch, enhancing both flexibility and efficiency in developing effective DRL applications for energy distribution networks.

In summary, this thesis addresses key challenges in enforcing operational constraints, enhancing learning efficiency, and reducing computational costs in DRL-based optimal DNs operation. The developed algorithms and frameworks significantly advance the reliability, safety, and practicality of DRL applications in modern power systems, paving the way for more effective and accurate distribution network operation and management.

SAMENVATTING

De integratie van gedistribueerde energiebronnen (DER's) en de toenemende penetratie van hernieuwbare energieopwekking hebben de complexiteit en onzekerheid van moderne distributienetwerken aanzienlijk vergroot. Deze ontwikkelingen vereisen geavanceerde dispatch-algoritmen die in staat zijn om de variabiliteit en operationele beperkingen van dergelijke systemen te beheersen. Dit proefschrift richt zich op de ontwikkeling van modelvrije deep reinforcement learning (DRL)-algoritmen om een betrouwbare, veilige en kosteneffectieve werking van distributienetwerken (DN's) te waarborgen. De onderzoeksvragen in dit proefschrift behandelen verschillende uitdagingen met betrekking tot het afdwingen van operationele beperkingen, leerefficiëntie en het verminderen van computationele kosten bij DRL-gebaseerde optimale werking van DN's.

Ten eerste is de handhaving van vermogensbalansbeperkingen essentieel voor het behouden van stabiliteit en betrouwbaarheid van het systeem. Standaard modelvrije DRL-benaderingen hebben vaak moeite om operationele beperkingen strikt af te dwingen. Om dit aan te pakken, is het DRL-algoritme MIP-DQN ontwikkeld, dat alle operationele beperkingen strikt afdwingt in de actieruimte, waardoor haalbare dispatch in real-time wordt gewaarborgd. Door optimalisatievoortuitgang voor deep neural networks (DNN's) te benutten, waarmee deze als mixed-integer linear programming (MILP) formuleringen kunnen worden weergegeven, zorgt dit algoritme ervoor dat beperkingen zelfs tijdens online uitvoering worden nageleefd. Vergelijkende prestatietests met state-of-the-art DRL-algoritmen hebben de effectiviteit van MIP-DQN aangetoond bij het handhaven van de vermogensbalansbeperking.

Ten tweede verbetert de integratie van operationele beperkingen en bestaande kennis in DRL-frameworks de haalbaarheid van oplossingen en vermindert het de complexiteit van het probleem. Dit proefschrift introduceerde het DistFlow Safe Reinforcement Learning (DF-SRL)-algoritme, dat gebruik maakt van deskundige kennis om de relatie tussen de acties van de agent en spanningsvariaties in DN's nauwkeurig in kaart te brengen. Het DF-SRL-algoritme voegt een veiligheidslaag toe bovenop het DRL-beleid om potentieel onveilige acties te hercalibreren, waardoor spanningsbeperkingen strikt worden nageleefd tijdens zowel de trainings- als toepassingsfasen.

Ten derde vereist het aanpakken van sequentiële beperkingen in DRL-gebaseerde optimale DN's-werking dynamische aanpassing aan de timing en volgorde van acties om veilige en betrouwbare operaties te waarborgen. Voortbouwend op eerder werk werd een Mixed-Integer Programming Deep Reinforcement Learning (MIP-DRL)-framework ontwikkeld. Dit veelzijdige framework stelt verschillende standaard actor-critic DRL-algoritmen in staat om operationele beperkingen strikt af te dwingen. Door gebruik te maken van het robuuste vermogen van MIP om beperkingen af te dwingen, garandeert het MIP-DRL-framework nul schendingen van beperkingen tijdens online uitvoering, waardoor de toepasbaarheid van DRL in real-time toepassingen wordt uitgebreid.

Ten vierde is het waarborgen van een veilig en efficiënt algoritme essentieel voor

praktische implementatie. Een Safe Imitation Reinforcement Learning-framework dat Twin Delayed Deep Deterministic Policy Gradient (TD3) en imitatie learning (IL) combineert, is voorgesteld. Dit framework verbetert de prestaties en trainingsefficiëntie, terwijl operationele beperkingen strikt worden gehandhaafd. De offline trainingsfase gebruikt een dual-gradientstrategie met zowel een behavior cloning (BC) beleid als een critic-netwerk om de training te stabiliseren en het leerproces te versnellen. Een veiligheidslaag controleert de door TD3BC aanbevolen acties, filtert onveilige acties uit en garandeert operationele haalbaarheid in scenario's die niet door deskundige gegevens worden gedekt.

Ten slotte zijn het verminderen van computationele kosten en het versnellen van het leerproces essentieel voor de praktische implementatie van DRL-algoritmen in DN's. De RL-ADN-bibliotheek werd geïntroduceerd als een open-source tool specifiek ontworpen voor DRL-gebaseerde optimale ESS-operatie in distributienetwerken. RL-ADN biedt uitgebreide aanpassingsmogelijkheden en integreert een innovatieve data augmentatiemodule met Gaussian Mixture Models-Copula (GMC) en de Tensor Power Flow (TPF)-oplosser, waardoor de rekentijd voor load flow-berekeningen aanzienlijk wordt verminderd. Deze bibliotheek zet een nieuwe standaard in DRL gebaseerde ESS-dispatch en verbetert de flexibiliteit en efficiëntie bij het ontwikkelen van effectieve DRL toepassingen voor energiedistributienetwerken.

Kortom, dit proefschrift behandelt belangrijke uitdagingen bij het afdwingen van operationele beperkingen, het verbeteren van leerefficiëntie en het verminderen van computationele kosten bij DRL-gebaseerde optimale werking van DN's. De ontwikkelde algoritmen en frameworks dragen aanzienlijk bij aan de betrouwbaarheid, veiligheid en praktische toepasbaarheid van DRL-toepassingen in moderne energiesystemen en bereiden de weg voor effectievere en nauwkeurigere operationele planning en beheer van distributienetwerken.

1

INTRODUCTION

This introductory section provides a comprehensive overview of the research background, questions, and methodologies employed in this thesis. Energy systems are transitioning from fossil-based to renewable sources to combat climate change and ensure sustainable energy futures. This shift positions electricity as the dominant energy source in energy sectors. However, the integration of renewable sources introduces significant complexity and uncertainty, particularly in systems with high levels of distributed energy resources (DERs) penetration. This complexity necessitates advanced scheduling algorithms to maintain grid stability and ensure reliable operations. This thesis aims to develop model-free deep reinforcement learning (DRL) algorithms to address optimal distribution network operational challenges. This chapter introduces the research questions that steer the project and the approaches and contributions made by the study. To aid readers in navigating this thesis, the final segment of this chapter outlines the structure and discusses the content of each subsequent chapter comprehensively.

1

1.1. BACKGROUND AND MOTIVATION

The global transition towards renewable energy is a cornerstone of efforts to mitigate climate change and reduce the environmental impact of traditional energy systems [1]. With the 2015 Paris Agreement and subsequent international climate goals, there is a clear global commitment to accelerate the adoption of renewable energy sources like wind, solar, and energy storage systems (ESSs) [2]. These distributed energy resources (DERs) offer opportunity to create a more flexible, resilient, and sustainable energy system, crucial for meeting ambitious global climate targets and ensuring long-term environmental sustainability [3].

However, the rapid increase in DER integration, particularly intermittent resources such as solar photovoltaics (PV) and wind turbines, presents significant challenges for the operational management of distribution networks [4]. Without proper management, the variability in generation from renewable energy sources can lead to voltage fluctuations, frequency instability, and even power outages in severe cases [5]. Optimal distribution networks (DNs) operation plays a pivotal role in ensuring that electricity infrastructure can reliably integrate large shares of renewable energy. As countries work towards increasing the share of renewable energy in their overall energy mix, the challenges associated with managing variable power generation, storage, and demand also grow more complex [6]. DNs optimal operation refers to the optimization of energy generation, storage, and consumption in a way that meets both operational constraints and sustainability objectives, such as reducing carbon emissions and minimizing system cost. The importance of performing optimal DNs grows even more critical as renewable energy penetration increases [7]. Traditional power systems were designed around centralized, predictable energy sources like coal, gas, and nuclear power, usually connected at high voltage levels. These sources provide stable and controllable outputs, making it easier to schedule and balance the grid. In contrast, renewable energy sources are distributed and decentralized, with outputs that fluctuate based on weather conditions and other external factors [8]. This presents new challenges for maintaining grid stability and reliability. With higher penetration levels of renewables, DNs must be able to dynamically adjust to the variability in both generation and consumption. For instance, solar generation peaks during the day, while wind power may fluctuate depending on weather conditions. At the same time, demand patterns also vary, making it increasingly difficult to ensure that supply matches demand in real-time. The optimal DNs operation requires rapid, informed decisions that balance these factors, ensuring the system remains reliable while maximizing the use of renewable energy.

From a sustainability perspective, optimal DNs operation helps achieve several key objectives:

- Maximizing Renewable Energy Utilization: By optimally scheduling when and how much energy to generate, store, and consume, optimal DNs operation ensures that renewable energy resources are utilized as efficiently as possible. This reduces reliance on fossil fuels and lowers the carbon intensity of energy generation.

- Minimizing Curtailment: Renewable energy sources, especially solar and wind, can sometimes produce more power than the grid can handle. Without effective scheduling, this excess power may be wasted or curtailed. Optimal DNs operation

mitigates this issue by optimally dispatching excess energy to storage systems or redirecting it to other uses, thereby enhancing system flexibility.

- Reducing Greenhouse Gas Emissions: By improving the efficiency of energy generation and reducing the need for conventional power plants to ramp up during peak periods, optimal DNs operation directly contributes to lowering the overall carbon footprint of energy systems.

- Supporting Grid Resilience: As more renewable energy is integrated into the grid, the ability to maintain stable and reliable operations becomes more difficult. Optimal DNs operation contributes to grid stability by optimizing the use of energy storage and demand response to absorb fluctuations in renewable generation.

Without effective optimal DNs operation, renewable energy sources may not be fully exploited, and their potential to contribute to climate goals would be diminished. Moreover, poor scheduling could lead to increased reliance on backup generation from fossil fuels, negating the environmental benefits of renewables. Therefore, optimal DNs operation is not just a technical necessity but a key enabler of the transition to a sustainable, low-carbon energy system.

1.1.1. TRADITIONAL APPROACHES AND LIMITATIONS

Traditionally, model-based approaches have been used to address the optimal DNs operation problem. These methods rely on precise mathematical models to represent the operational constraints of the system, formulating the problem as linear, nonlinear, or dynamic programming tasks [9]. However, these approaches face significant limitations, particularly in real-time applications. The need for accurate models, coupled with the computational complexity of solving large-scale problems, makes these methods less effective as the complexity of the DNs increases [10]. Moreover, model-based methods typically handle uncertainty through probabilistic models or representative scenarios, leading to stochastic or robust optimization formulations [11]. While these approaches can produce good solutions under controlled conditions, they often struggle with scalability and require simplifications that reduce their effectiveness in dynamic, real-world environments. The computational burden associated with these methods further limits their practicality for real-time decision-making.

1.1.2. THE EMERGENCE OF REINFORCEMENT LEARNING APPROACHES

To overcome the limitations of model-based methods, model-free approaches, particularly those based on reinforcement learning (RL) have gained prominence [12]. RL algorithms model the decision-making process as a Markov Decision Process (MDP), enabling agents to learn the dynamics of the system through interaction with the environment [13]. This interaction allows RL to capture the inherent uncertainty and complexity of DNs, especially those influenced by renewable energy sources, whose variability poses significant challenges to traditional optimization methods.

Model-free RL methods emerged as a solution to overcome the limitations of model-based approaches, which rely on precise mathematical formulations of the system dynamics and operational constraints. In contrast, model-free RL methods are trained in

simulators and then applied to real world systems [14]. The RL agent explores the simulator, gradually improving its policy by trial and error, using rewards as an indicator of performance. This approach is particularly useful for DNs operational problems, where uncertainties, such as the variability of solar and wind power generation, make it difficult to rely solely on predefined models.

One of the foundational approaches is value-based methods, such as Q-learning and SARSA, which focus on estimating the value of state-action pairs. The goal is to learn the value function that represents the expected cumulative reward from any given state-action pair [15]. Over time, these methods have been enhanced by both linear and non-linear function approximations, which led to the development of more sophisticated algorithms such as Deep Q-Networks (DQN) [16]. DQN leverages deep neural networks (DNNs) to approximate the value function in high-dimensional state spaces, making it particularly suitable for complex, large-scale problems. DQN and its variants, such as Double DQN [17], Dueling DQN [18], and Rainbow DQN [19], address issues like overestimation bias and instability, which are common in value-based methods. Techniques like prioritized experience replay and distributional learning have also been integrated to improve the performance and robustness of these algorithms [20]. Despite these advancements, value-based methods still face challenges when applied to the optimal operation of DNs with continuous action spaces (e.g., battery charging/discharging levels or distributed generation outputs), where exhaustive action space exploration becomes computationally prohibitive.

To address the limitations of value-based methods in continuous action spaces, policy based methods have gained attention. These methods focus directly on optimizing the policy function that maps states to actions, instead of estimating value functions. The Policy Gradient (PG) method is one of the foundational approaches, which updates the policy in the direction that maximizes the expected cumulative reward [21]. However, PG methods can suffer from high variance and instability. Advanced policy-based methods like Trust Region Policy Optimization (TRPO) and Proximal Policy Optimization (PPO) have been developed to address these issues by ensuring stable updates and preventing the policy from making large, untrustworthy jumps [22]. These methods are especially well-suited for DNs operation problems, where small changes in policy can lead to large swings in system performance due to the interconnected nature of distribution networks. The Actor-Critic framework, including algorithms like Advantage Actor-Critic (A2C) and Asynchronous Advantage Actor-Critic (A3C), combines the strengths of value-based and policy-based approaches. In this framework, the actor selects actions based on the current policy, while the critic evaluates the chosen actions by estimating the value function. This combination allows for more efficient learning and greater stability, making it a strong candidate for solving optimal DNs operational problems where both optimal decision-making and long-term evaluation are critical.

For problems with continuous action spaces, Deterministic Policy Gradient (DPG) methods have become highly effective. These methods, such as Deep Deterministic Policy Gradient (DDPG), extend policy gradient approaches to continuous spaces by using a deterministic policy function rather than a stochastic one [23]. DDPG can handle high-dimensional action spaces without requiring the agent to sample across the entire action space making it a powerful tool for DNs operation problems involving decisions such as

energy storage dispatch or renewable generation control. Enhanced versions of DDPG, like Twin Delayed DDPG (TD3) [24] and Distributed Distributional DDPG (D4PG), address common issues such as overestimation bias by introducing techniques like the delayed policy update and distributional learning. These enhancements are particularly important for ensuring that the learned policies are not only optimal but also safe and reliable, which is critical in optimal DNs operation where operational constraints must be respected.

1.1.3. CHALLENGES

DRL algorithms have been successfully applied to various DNs operations; tasks, including home energy management, microgrid operation [25], and DNs control [26]. These applications demonstrate the ability of DRL to handle uncertainty and complexity by learning directly from historical data and simulations [27].

Despite its potential, applying DRL to the optimal operation of DNs introduces several critical challenges, which can be broadly categorized as follows:

- Constraint Enforcement: Ensuring that DRL-based solutions respect real-world operational constraints, such as power balance and voltage magnitude limits, is essential to avoid system failures.
- Data Efficiency and Uncertainty: DRL requires vast amounts of historical and simulated data for training. However, generating and accessing this data, especially for rare events or corner cases, is challenging.
- Computational Efficiency: The computational cost of training DRL agents for large-scale, real-time systems is prohibitive, often requiring substantial time and computing resources.

1.2. RESEARCH OBJECTIVE AND QUESTIONS

This thesis focuses on developing model-free DRL algorithms, aiming to ensure reliable, safe, cost-effective, and sustainable operations.The specific challenges and corresponding research questions are elaborated in this section.

1.2.1. ENFORCING DISTRIBUTION NETWORKS OPERATIONAL CONSTRAINTS USING DRL

The operation of distribution networks (DNs) is governed by a set of strict operational constraints to avoid voltage magnitude problems and congestion issues. Failure to adhere to these constraints can lead to severe operational risks, system failures, or even cascading outages. Thus, ensuring safety is paramount, and it is typically achieved by enforcing these operational constraints during the dispatch process [28].

In the context of the optimal DNs operation problem formulated as a Markov Decision Process (MDP), maintaining safety means that any decision or action taken by an RL agent must respect these operational constraints, particularly the power balance between supply and demand. The power balance constraint ensures that, at every time step, the total energy generated and stored must match the energy consumed plus any

system losses. However, due to the inherent stochasticity and complexity of real-world power systems, maintaining this balance can be challenging when relying on data-driven approaches like RL, which do not explicitly enforce such constraints in their formulation.

Traditional DRL algorithms, while capable of optimizing system performance in terms of minimizing costs or maximizing efficiency, often lack formal safety guarantees [29]. These algorithms are designed to maximize rewards based on learned experiences without being directly aware of physical constraints, such as power balance, ramping limits, and grid reliability requirements. To address this, practitioners often introduce penalty terms in the reward function to incentivize the agent to avoid actions that violate these constraints. For example, when a DRL agent's action results in power imbalance, a penalty is applied to the reward function to discourage the agent from repeating such actions. However, this approach does not explicitly prevent the agent from making unsafe decisions, especially under extreme conditions such as high variability in renewable generation or sudden load changes.

Our research [30] demonstrates that while penalty-based methods can guide the learning process to reduce operational costs, they are insufficient in ensuring the feasibility of actions, particularly in scenarios involving extreme system conditions. In these cases, the DRL agent may still violate critical constraints, leading to power imbalances and potential operational instability. For instance, when the system is under stress due to sudden spikes in demand or generation variability, penalty terms alone may not be sufficient to prevent the agent from selecting actions that cause power imbalances, as the penalties may not accurately reflect the severity of constraint violations under these conditions.

Therefore, ensuring safety in the optimal DNs operation formulated MDP requires more than just penalizing constraint violations; it necessitates a framework where operating constraints—such as power balance, ramping limits, and capacity restrictions—are rigorously enforced throughout the learning process. Thus, our first research question is formulated as follows: **Research Question 1: How to enforce distribution networks operational constraints using DRL?**

1.2.2. LEVERAGING DOMAIN KNOWLEDGE TO ENSURE SAFETY, INCREASE PERFORMANCE, AND ENHANCE COMPUTATIONAL EFFICIENCY IN DRL FOR OPTIMAL DNS OPERATION

Despite the advancements in model-free RL algorithms, DRL applications in distribution networks (DNs) often remain constrained by their model-free nature. Typically, model-free DRL requires a considerable amount of time and numerous interactions with the environment to learn even basic operational strategies. This reliance on extensive environment interaction often results in long training times and substantial computational costs, which limit the practical applicability of DRL in real-time DN operations.

In practice, however, there exists a wealth of domain knowledge that can aid this process by enhancing both the safety and efficiency of DRL. Domain knowledge—such as insights into power system behavior, known operational constraints, and grid reliability requirements—can be incorporated into the RL framework to guide the learning process and reduce the agent's dependence on raw environment interactions. By embedding relevant domain knowledge into the DRL framework, it is possible to not only accelerate learning but also improve the overall safety and performance of the algorithm

when applied to complex DNs operational problems.

The objective of this research question is to investigate methods for integrating domain knowledge within DRL algorithms to optimize the operation of DNs, specifically focusing on:

- Enhancing Safety: Ensuring that the agent's actions consistently respect critical operational constraints, thereby maintaining system reliability.
- Increasing Performance: Improving the decision-making capabilities of the agent to achieve efficient and cost-effective DNs operations.
- Enhancing Computational Efficiency: Reducing the amount of environment interaction needed during training, which in turn minimizes computational cost and expedites convergence.

This leads us to our second research question: **Research Question 2: How can domain knowledge be leveraged to ensure safety, increase performance, and enhance computational efficiency in DRL for optimal DNs operation?**

1.2.3. REDUCING COMPUTATIONAL COST AND ACCELERATING TRAINING FOR DRL BASED OPTIMAL DNS OPERATIONAL PROBLEMS

To train an effective DRL agent capable of optimally operation a DN, it is essential not only to develop advanced DRL algorithms but also to have a well-designed training environment. Currently, existing environments, such as PowerGridWorld and Grid2OP, exhibit limitations that restrict their applicability for diverse DN operational challenges. Specifically, these environments often lack sufficient customizability, computational efficiency, and data augmentation capabilities, which are critical for training robust DRL agents.

Firstly, customizability is a crucial requirement in environment design. DNs operational problems encompass various objective functions, decision variables, and unique problem characteristics. Existing environments are typically tailored to specific, limited operational scenarios and often lack the flexibility to adapt to different DN configurations and objectives. This restricts the training environment's ability to simulate a wide range of realistic scenarios, hindering the generalizability of the trained agents.

Secondly, computational efficiency poses a significant challenge in DRL for DN operations. Interactions between the agent and the environment often involve extensive power flow calculations, which are integral to the learning process but can be computationally intensive. Training an agent to convergence requires millions of environment interactions, resulting in a bottleneck due to the repeated need for time-consuming power flow computations. Existing environments, built on standard iterative power flow calculation methods such as Newton-Raphson, exacerbate this issue, as they are not optimized for high-frequency interaction with DRL algorithms.

Finally, data augmentation is vital for enhancing the diversity of training data. Leveraging DRL to develop optimal dispatch strategies relies on training with diverse historical data, particularly in scenarios with variable renewable generation, fluctuating load profiles, and changing price signals. However, historical data specific to DN configurations is often scarce, limiting the scope and performance of the trained agents. Data

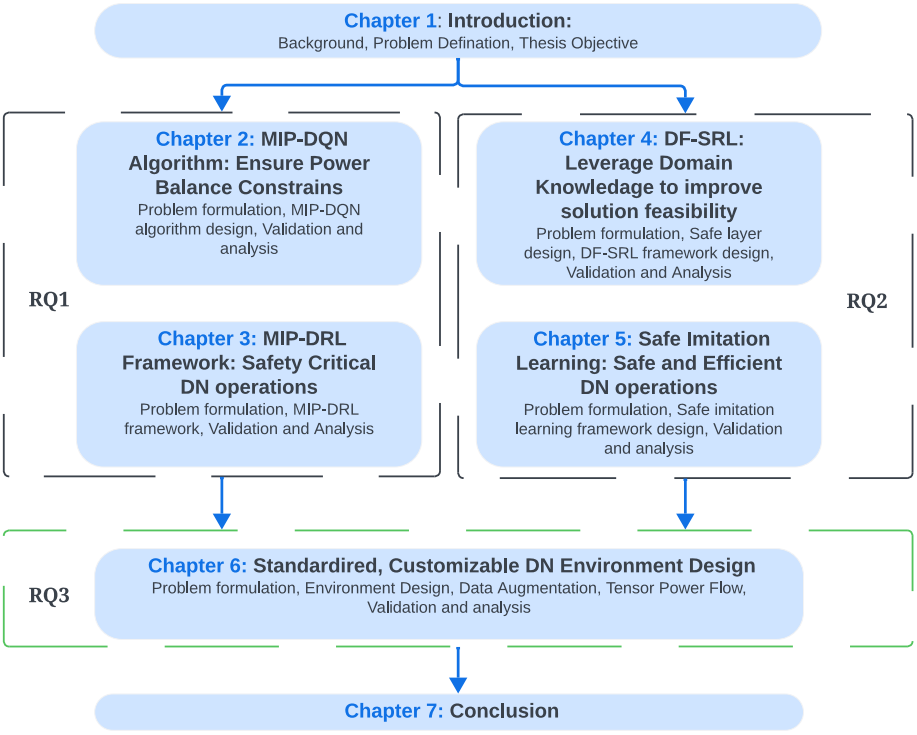


Figure 1.1: Framework diagram for the thesis

augmentation techniques can help bridge this gap by synthetically expanding the training dataset, thereby improving the robustness and effectiveness of DRL agents under varying conditions.

Based on these challenges, our third research question is formulated as follows: **Research Question 3: How to design a standardized, customizable, and data-augmented environment to reduce computational cost and accelerate DRL training for optimal DN operational problems?**

1.3. CONTRIBUTIONS AND THESIS OUTLINE

This thesis makes several significant contributions to the field of DRL-based optimal DNs operation by addressing the critical challenges of enforcing operational constraints, ensuring safety, and improving computational efficiency. The framework of this thesis is shown in Fig. 1.1 and contributions are summarized as follows:

Chapter 1 introduces the research problem, objectives, and research questions that set the stage for the subsequent chapters. This chapter outlines the complexity of the optimal DNs operation problems due to the massive integration of renewable-based resources and the inherent uncertainty of the system.

Chapter 2 addresses *Research Question 1: How to enforce distribution network opera-*

tional constraints using DRL? Ensuring power balance and adhering to operational constraints is essential for maintaining system stability and reliability. Traditional model-based approaches face limitations due to their dependence on accurate models and high computational complexity, while DRL algorithms, despite their potential, often struggle to enforce these constraints directly. To address this, we proposed the MIP-DQN framework, which leverages recent advances in optimization to represent Deep Neural Networks (DNNs) as mixed-integer linear programming (MIP) formulations. This enables the DRL algorithm to strictly enforce all operational constraints in the action space, ensuring feasible, real-time scheduling. By testing the MIP-DQN algorithm against state-of-the-art DRL algorithms such as DDPG, PPO, SAC, and TD3, we demonstrate its effectiveness in generating feasible and optimal schedules that satisfy power balance constraints.

Chapter 3 further investigates *Research Question 1* by introducing the MIP-DRL framework. This framework extends the applicability of DRL algorithms to more complex and safety-critical DN operations by ensuring strict adherence to operational constraints during real-time execution in distribution networks. MIP-DRL incorporates additional network constraints, enhancing the algorithm's ability to handle complex distribution network scenarios safely and reliably.

Chapter 4 focuses on *Research Question 2: How to leverage domain knowledge to ensure safety, increase performance, and enhance computational efficiency in DRL for optimal DN operation?* Integrating operational constraints into DRL while leveraging prior domain knowledge helps reduce problem complexity and improve solution feasibility. In this chapter, we introduce the DistFlow Safe Reinforcement Learning (DF-SRL) algorithm, which incorporates expert insights to establish a robust relationship between the agent's actions and voltage magnitude variations in distribution networks. This algorithm overlays a safety layer atop the DRL policy, recalibrating potentially unsafe actions to conform to safe operational parameters. DF-SRL thus ensures that voltage magnitude constraints are strictly enforced during both training and deployment.

Chapter 5 further investigates*Research Question 2*, addressing the challenge of designing a safe and efficient DRL framework for DN operations. To achieve this, we proposed a Safe Imitation Reinforcement Learning (IRL) Framework that combines Twin Delayed Deep Deterministic Policy Gradient (TD3) and Inverse Reinforcement Learning (IRL). This framework not only enhances training efficiency but also rigorously enforces operational constraints. During offline training, a dual-gradient strategy is employed, utilizing both the behavior cloning (BC) policy and the critic network. This approach stabilizes the training process and expedites learning, addressing the computational and exploration challenges associated with traditional TD3. Additionally, a safety layer filters out unsafe actions recommended by the trained TD3BC algorithm, redirecting them into safer alternatives to ensure operational feasibility in scenarios not covered by expert data.

Chapter 6 addresses *Research Question 3: How to design a standardized, customizable, and data-augmented environment to reduce computational cost and accelerate DRL training for optimal DN operational problems?* Training DRL agents in distribution networks is computationally intensive, often requiring extensive power flow calculations and diverse data. This chapter introduces RL-ADN, an open-source library for DRL-

based optimal ESS dispatch in distribution networks. RL-ADN provides flexibility in environment design by supporting a broad range of DN configurations and objectives, extending to the modeling of DN topologies, integration of ESSs, and customization of research objectives. It also includes a novel data augmentation module based on Gaussian Mixture Models-Copula (GMC), enhancing training data diversity and improving DRL agent performance. To address computational challenges, RL-ADN incorporates the TPF solver, which reduces power flow calculation time by tenfold without sacrificing accuracy [31]. Additionally, RL-ADN offers four state-of-the-art DRL algorithms and a model-based approach with perfect forecasts as a baseline. RL-ADN thus sets a new standard in DRL-based ESS dispatch with its innovative features, flexibility, and efficiency, facilitating more effective and accurate DRL applications in energy distribution networks..

Finally, **Chapter 7** concludes the thesis by summarizing the key findings and proposing future research directions. This chapter highlights the contributions made in this thesis and suggests potential areas for further exploration to advance the field of DRL-based energy system optimization.

2

Optimal Energy System Scheduling Using A Constraint-Aware Reinforcement Learning Algorithm

The massive integration of renewable-based distributed energy resources (DERs) inherently increases the energy system's complexity, especially when it comes to defining its operational schedule. Deep reinforcement learning (DRL) algorithms arise as a promising solution due to their data-driven and model-free features. However, current DRL algorithms fail to enforce rigorous operational constraints (e.g., power balance, ramping up or down constraints) limiting their implementation in real systems. This chapter proposed a DRL algorithm namely, MIP-DQN, designed to enforce all operational constraints in the action space, ensuring the feasibility of the defined schedules in real-time distributed energy system operation. This is done by leveraging recent optimization advances for deep neural networks (DNNs) that allow their representation as a MIP formulation, enabling further consideration of any action space constraints. Comprehensive numerical simulations show that the proposed algorithm outperforms existing state-of-the-art DRL algorithms, obtaining a lower error when compared with the optimal global solution (upper boundary) obtained after solving a mathematical programming formulation with perfect forecast information; while strictly enforcing all operational constraints (even in unseen test days).

Parts of this chapter have been published in the International Journal of Electrical Power & Energy Systems with the title: *Optimal Energy System Scheduling Using A Constraint-Aware Reinforcement Learning Algorithm.* doi:10.1016/j.ijepes.2023.109230 [32].

2.1. Introduction

To reduce the impact of the energy sector on the environment, distributed energy resources (DERs) are being integrated into our energy systems. Such DERs, in the form of renewable-based systems (e.g., PV systems and wind turbines) and small-scale energy storage systems (ESSs), provide more flexibility, enabling a more efficient operation. Nevertheless, these DERs also increase the energy system's complexity, especially when it comes to defining its operational schedule. Moreover, due to their weather-dependent nature, renewable-based DERs inherently increase the energy system's levels of uncertainty, requiring scheduling algorithms capable of providing fast and good-quality, but feasible, solutions [33]. In the technical literature, two main approaches are available to deal with the optimal scheduling of energy systems; namely, *model-based* and *model-free* approach. A detailed literature review is presented next.

2.1.1. Literature Review

In general, model-based approaches rely on precise models to build complex mathematical formulations to consider the energy system's operational constraints. Depending on how these constraints are modeled, the derived mathematical formulations can be classified as linear, nonlinear programming, or dynamic programming problems [34]. In this regard, in [35], a mixed-integer nonlinear programming (MINLP) formulation is used to determine the optimal operation of an unbalanced three-phase energy system. In order to reduce the complexity of the proposed formulations, linearizations and simplifications are introduced. Similar work has been done in [36]. Nevertheless, the model-based nature of these methods requires considerable precision of the built mathematical models, which limits their performance, especially if uncertainty is to be considered.

Generally, in model-based approaches, uncertainty is modeled either by using a probability distribution function or by leveraging a set of representative scenarios, leading to stochastic or robust mathematical formulations, such as the ones presented in [37, 38, 39]. Other approaches, such as the one in [40], leverage a rolling time horizon approach to eliminate the forecast error when defining the DERs optimal energy scheduling. To guarantee the feasibility of the defined schedule under various operational scenarios, in [41], an adjustable two-stage robust optimization framework is proposed, solving simultaneously a day-ahead scheduling and real-time regulation problem of an integrated energy system. In [42], a chance-constrained programming model is proposed to schedule an active distribution network incorporating office buildings. Nevertheless, modeling the probability distribution of uncertain data is challenging, while using a large number of scenarios may cause a computational burden. Therefore, although capable of providing good quality solutions, existing model-based approaches are not adequate for handling the increased uncertainty level of renewable-based energy systems, as their performance and efficiency mainly depend on the accuracy of the used models and their approximations. Moreover, the computational complexity of these methods increases dramatically with the system size, imposing scalability and convergence challenges.

To overcome this, model-free approaches have been introduced as an alternative solution. The most promising approach is based on the use of reinforcement learning (RL) [43], modeling the decision-making problem as a Markov Decision Process (MDP). One of the most interesting features of RL algorithms is that they can learn any system's

dynamics by interaction, providing good-quality solutions guided by a reward value used as a performance indicator [44]. Recently, deep reinforcement learning (DRL) algorithms have shown good performance when solving MDPs in energy systems tasks [45], ranging from, home energy management [46], microgrid dispatch [47], and electricity network operation [48]. Other applications include, for instance, a standarized DRL approaches for demand response in smart buildings [49], and learning to solve fast optimal power flow problems using DRL algorithms, specifically the proximal policy optimization (PPO) algorithm and imitation learning [50]. In [51], a performance comparison of the soft actor-critic (SAC) algorithm with a rule-based control method on the surrogate simulation model developed by [52], is presented. In [26], the voltage regulation problem of a distribution network is first modeled as a partial-observable MDP, and then multi-agent DRL algorithms are leveraged to execute the optimal solutions. In [53], a DRL approach-based proactive operation framework is proposed to model the stochastic behavior and uncertainty of solar energy for residential buildings. In [54], a DRL algorithm is developed to solve a stochastic energy management problem considering power flow constraints, resulting in an optimal policy that minimizes total operational cost (although operational constraints are disregarded).

Different from the energy-related MDPs presented above, the operational schedule of DERs within an energy system must enforce a rigorous set of operational constraints to ensure a reliable and safe operation, e.g., generation and consumption must always be balanced during real-time operation, ramping-up and down constraints, etc. Nevertheless, current DRL algorithms lack of safety guarantees [55], as these constraints cannot be directly imposed in the algorithm's formulation. Different strategies to indirectly enforce operational constraints have been proposed to overcome this. In [56], a DG unit is set as a slack bus with unlimited generation capacity, avoiding unbalance by the outputs of the generators controlled by DRL agents. In [57], a penalty term is added to the reward function to guide the learning process aiming to reduce operating costs while enforcing power balance. A similar penalty approach has been used to enforce voltage magnitude constraints in case the electricity network operation is considered. For instance, [58] modeled the dispatch of PV inverters as an MDP, and built a decentralized dispatch framework penalizing RL agents when actions lead to voltage violations. In research [59], an on-policy RL algorithm with eligibility traces is developed to dispatch the energy storage system to minimize the cost and regulate voltage magnitudes. A similar work is presented in [60]. In [61], a service assistant restoration problem is modeled as MDP. Then, imitation learning is employed as expert demonstrations enabling a deep deterministic policy gradient (DDPG) agent learn a safe policy for online implementation. In [62], a double auction market-based coordination framework is proposed to schedule the energy trading between multi-energy microgrids. Multi-agent twin delayed deep deterministic algorithm (TD3) is used to solve the formulated problem, while a large penalty is imposed on the reward function to reduce the energy unbalance. In [63], the soft-actor critic (SAC) algorithm is leveraged to control a virtual power plant to provide frequency regulation services, penalizing any frequency deviation. Nevertheless, although these strategies may enforce operational constraints during training, they are either based on nonpractical assumptions or fail to guarantee the feasibility of the defined operating schedule in real-time, especially in cases of large peak consumption or

renewable-based generation [30].

Strategies based on safe RL have also been proposed to directly enforce operational constraints, exploiting results from different research areas, such as robot manipulation [64, 65]. In recent years, considerable attention has been given to combining verification techniques with reinforcement learning (RL) to enforce safety constraints explicitly, a sub-field often referred to as Shielded Reinforcement Learning (Shielded RL) [66, 67, 68]. Shielded RL typically introduces a "shield" layer between the RL agent and the environment, employing formal verification or optimization techniques to prevent unsafe actions proactively, thus guaranteeing constraint satisfaction during exploration and execution [66]. In [67], the shielding framework is extend to multi-agent settings, emphasizing scalability and formal safety guarantees in complex interaction scenarios. In [69], the Worst-Case Soft Actor Critic algorithm utilized a distributional safety critic and Conditional Value-at-Risk constraints to optimize policies under worst-case scenarios, thus providing stronger safety assurances in continuous action domains. In [70], an action projection layer is implemented, correcting the action defined by the DRL algorithm via a projection operator. Unfortunately, this projection operator degrades the DRL algorithm's performance, as shown in [71]. In [72], safe DDPG is used for real-time automatic control of a smart hub, while a safety net is used to estimate the feasibility of decided actions. A similar strategy is proposed in [73], in which the action proposed by the DRL algorithm is used as starting point to solve a mathematical programming formulation, ensuring constraints compliance. In [74], a constrained policy gradient approach is proposed, updating the parameters of the DNN model in the direction that minimizes the power unbalance. In [75], the same approach is used to solve an EVs coordination problem. This policy optimization approach allows the DRL algorithm to provide a probabilistic notion of safety.

Nevertheless, feasibility is paramount in energy systems operation, and it should be certifiable. In this regard, enforcing operational constraints during the online scheduling stage is a critical challenge for DRL algorithms and it must be addressed to enable their wide adoption in real systems. A summary of the discussed research literature is presented in Table 2.1. The openness and free online availability of the algorithms discussed here are also highlighted in Table 2.1.

2.1.2. CONTRIBUTIONS

To overcome the above-discussed limitations, this paper proposes a DRL algorithm, MIP-DQN, to define the optimal schedule of a renewable-based energy system, capable of *strictly* enforcing all the operational constraints in the action space, ensuring the feasibility of the defined scheduled in real-time operation. To do this, we used recent optimization advances for DNNs that allow their representation as a mixed-integer linear (MIP) formulation, enabling further consideration of any action space constraints. Such approaches have been also employed in the context of feature visualization and adversarial machine learning [76]. The performance of the proposed algorithm has been compared with other state-of-the-art DRL algorithms available in the literature, including DDPG [23], PPO [22], SAC [77], and TD3 [24] algorithms, to show its effectiveness. A comparison with the optimal global solution is also presented, obtained by solving the energy system scheduling problem as a mathematical programming formulation con-

sidering full knowledge of future information (i.e., consumption, dynamic prices, and renewable-based generation). The main contributions of this paper are as follows:

- A value-based DRL algorithm to solve the energy system scheduling problem is proposed, capable of dealing with continuous action spaces. Different from other actor-critic DRL algorithms (e.g., DDPG, PPO, and TD3), we make use of the action-value function approximated using a DNN, while discarding the policy model learning used during exploration.
- An innovative online execution approach that guarantees that the proposed DRL algorithm *strictly* meets all operational conditions in the action space (e.g., the power balance constraint), even in unseen test data, is also proposed. This is done by leveraging new optimization results from DNNs that allow their representation as a MIP formulation, enabling further consideration of any action space constraints.

The rest of this paper is organized as follows. In Section 2.2, the optimal energy system scheduling problem is formulated. Then, in Section 2.3, the formulated problem is modeled as MDP while the proposed MIP-DQN algorithm is illustrated and used to solve the optimal energy system scheduling problem in Section 2.4. Simulation tests are presented, analyzed and discussed in Section 2.5, while conclusions are presented in Section 2.6.

2.2. MATHEMATICAL PROGRAMMING FORMULATION OF THE ENERGY SYSTEMS SCHEDULING PROBLEM

The structure of the considered energy system is shown in Fig. 2.1, including various DERs, such as solar photovoltaic (PV), ESSs, DGs, and loads, while a connection to the utility grid is leveraged to address a demand surplus or shortage problem. For tractable analysis, we assume the day-ahead market where the electricity price of each hour is revealed beforehand. For the energy system in Fig. 2.1, the optimal energy system scheduling problem can be modeled by the nonlinear programming (NLP) formulation described by (2.1)-(2.11). The objective function in (2.1) aims at minimizing the operating cost for the whole time horizon $\mathcal{T}$, comprising the operating cost of the DG units, as presented in (2.2), and the cost of buying/selling electricity from/to the main network, as in (2.3). Given the output power of DG units $P^G_{i,t}$, the operating cost can be estimated by using a quadratic function as in (2.2). The transaction cost between the energy system and the network is settled according to Time-of-Use (ToU) prices, in which it is assumed that selling prices are lower than the purchasing prices. In (2.3), ρ_t is the ToU price at time slot t, while P^N_t refers to the exported/imported power transaction to/from the network.

$$\min_{P^G_{i,t},P^B_{j,t}} \left\{ \sum_{t\in\mathcal{T}} \sum_{i\in\mathcal{G}} \left[C^G_{i,t}(\cdot) + C^E_t(\cdot) \right] \Delta t \right\}, \tag{2.1}$$

$$C^G_{i,t} = a_i \left(P^G_{i,t} \right)^2 + b_i P^G_{i,t} + c_i, \quad \forall i \in \mathcal{G}. \tag{2.2}$$

$$C^E_t = \begin{cases} \rho_t P^N_t & P^N_t > 0, \\ \beta \rho_t P^N_t & P^N_t < 0. \end{cases} \tag{2.3}$$

Table 2.1: Summary of research literature for DRL algorithms and constraint enforcing approaches.

Work	Research Problem	Constraint Enforcing	Advantages	Disadvantages	Open-access
[52]	Residential building energy schedule	Constraints disregarded	Simple	Not realistic	Yes
[46]	Microgrid operation				Yes
[53]	Residential buildings energy schedule				No
[25]	Microgrid operation	Penalty function	Easy to implement	No constraint guarantee	No
[26]	Voltage regulation				Yes
[50]	Optimal power flow				No
[54]	Energy dispatch				No
[78]	Energy dispatch				No
[30]	Optimal energy system scheduling				Yes
[58]	PV-inverter voltage regulation				No
[59]	Battery schedule and voltage regulation				No
[62]	Energy trading between microgrids				No
[61]	Restoration services	Imitation learning and penalty function	Accelerating training speed Improve the performance	No constraint guarantee	No
[56]	Energy Management	Unlimited slack bus	Simple	Not realistic	Yes
[70]	Energy management	Safe layer	Guarantee the feasibility	Performance deterioration Not fully model-free	No
[72]	Energy hub trading	Gaussian process			
		Safe layer			
		Action projection			
[73]	Microgrid operation				
[74]	Distribution network operation	Constrained policy optimization	Probabilistic guarantee feasibility	No constraint guarantee Higher computation time	No
[75]	EV management				

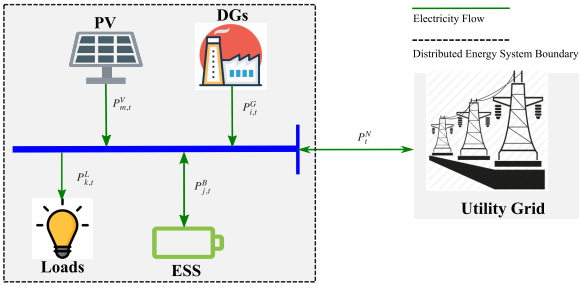


Figure 2.1: Illustration of the considered energy system structure composed of various DERs, such as solar photovoltaic (PV), ESSs, DGs, and loads.

Subject to:

$$\sum_{i\in\mathcal{G}} P^G_{i,t} + \sum_{m\in\mathcal{V}} P^V_{m,t} + P^N_t + \sum_{j\in\mathcal{B}} P^B_{j,t} = \sum_{k\in\mathcal{L}} P^L_{k,t} \qquad \forall t \in \mathcal{T} \quad (2.4)$$

$$\underline{P}^G_i \le P^G_{i,t} \le \overline{P}^G_i \qquad \forall i \in \mathcal{G}, \forall t \in \mathcal{T} \quad (2.5)$$

$$P^G_{i,t} - P^G_{i,t-1} \le RU_i \qquad \forall i \in \mathcal{G}, \forall t \in \mathcal{T} \quad (2.6)$$

$$P^G_{i,t} - P^G_{i,t+1} \le RD_i \qquad \forall i \in \mathcal{G}, \forall t \in \mathcal{T} \quad (2.7)$$

$$-\underline{P}^B_j \le P^B_{j,t} \le \overline{P}^B_j \qquad \forall j \in \mathcal{B}, \forall t \in \mathcal{T} \quad (2.8)$$

$$SOC^B_{j,t} = SOC^B_{j,t-1} + \eta_B P^B_{j,t} \Delta t / E^B_j \qquad \forall j \in \mathcal{B}, \forall t \in \mathcal{T} \quad (2.9)$$

$$\underline{SOC}^B_j \le SOC^B_{j,t} \le \overline{SOC}^B_j \qquad \forall j \in \mathcal{B}, \forall t \in \mathcal{T} \quad (2.10)$$

$$-\overline{P}^C \le P^N_t \le \overline{P}^C \qquad \forall t \in \mathcal{T} \quad (2.11)$$

Expression (2.4) defines the power balance constraint. Expression (2.5) defines the DG units generation power limits while (2.6) and (2.7) enforce the DG unit's ramping up and down constraints, respectively. Energy storage systems (ESSs) are modeled using (2.8)-(2.10). In this model, the operation cost of ESSs is not considered, while ESSs are allowed to schedule their discharge and charge power in advance. Expression (2.8) defines the charging and discharging power limits, while expression (2.9) models the state of charge (SOC) as a function of the charging and discharging power. Expression in (2.10) limits the energy stored in the ESSs, avoiding the impacts caused by over-charging and over-discharging. Finally, the main network export/import power limit is modeled by the expression in (2.11). Notice that to solve the mathematical formulation described by (2.1)-(2.11), *full knowledge* of future information (e.g., renewable-based generation, consumption and dynamic prices) is required, for instance, provided via a forecasting algorithm. The proposed DRL algorithm is able to provide good-quality solutions with only current information, as shown later. Next, the MDP formulation of the optimal scheduling problem is presented.

2.3. MDP Formulation & Value-Based DRL

The above-presented decision-making problem can be modelled as a finite MDP, represented by a 5-tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ represents the set of system states, $\mathcal{A}$ the set of actions, $\mathcal{P}$ the state transition probability function, $\mathcal{R}$ the reward function, and γ a discount factor. In this formulation, the energy system operator can be modeled as an RL agent. The state information provides an important basis for the operator to dispatch units. We define a state at time t as $s_t = (P_t^V, P_t^L, P_{t-1}^G, SOC_t)$, $\quad s_t \in \mathcal{S}$, while the actions, defining the scheduling of the DG units and the ESSs, as $a_t = (P_{i,t}^G, P_t^B)$, $\quad a_t \in \mathcal{A}$. Notice that the RL agent does not directly control the transaction between the energy system and the main network (i.e., P_t^N). Instead, after any action is executed, power is exported/imported from the main network to maintain the power balance. Nevertheless, a maximum power capacity constraint exists and must be enforced i.e., (2.11). Notice that if the maximum export/import limits are defined to be a low value (as done in this paper), in most cases, the power balance constraint will not be automatically met.

Given the state s_t and action a_t at time step t, the energy system transit to the next state s_{t+1} defined by the next transition probability function

$$p(S_{t+1}, R_t | S_t, A_t) = \Pr\{S_{t+1} = s_{t+1}, R_t = r_t \mid S_t = s_t, A_t = a_t\}, \tag{2.12}$$

which models the energy system's dynamics. In model-based algorithms, the uncertainty is predicted by a determined value or sampling from a prior probability distribution. In contrast, DRL algorithms are a model-free approach, capable of learning such dynamics from interactions. To guide learning, a reward r_t must be provided by the environment in order for the RL agent to quantify the goodness of any action taken. In the energy system scheduling problem, the reward function $\mathcal{R}(s_t, a_t)$ should guide the RL agent to take actions that minimize the total operating cost, while enforcing the power balance constraint. This can be done by using the reward function

$$\mathcal{R}_t(s_t, a_t) = r_t = -\sigma_1 \left[\sum_{i \in \mathcal{G}} \left(C_{i,t}^G + C_t^E \right) \right] - \sigma_2 \Delta P_t, \forall\, t \in \mathcal{T}, \tag{2.13}$$

in which ΔP_t corresponds to the power unbalance at time-step t, defined as,

$$\Delta P_t = \left| \sum_{i \in \mathcal{G}} P_{i,t}^G + \sum_{m \in \mathcal{V}} P_{m,t}^V + P_t^N + \sum_{j \in \mathcal{B}} P_{j,t}^B - \sum_{k \in \mathcal{L}} P_{k,t}^L \right|. \tag{2.14}$$

In (2.13), σ_1 and σ_2 are used to control the order of magnitude and the trade-off between the operating cost minimization and the penalty incurred in case of power unbalance. The procedure used to solve the proposed MDP using value-based RL algorithms is presented next.

2.3.1. DRL Value-Based Algorithms

Define $Q_\pi(S_t, A_t)$ as the action-value function that estimates the *expected cumulative reward* given that action a_t is taken at state s_t and following policy $\pi(\cdot)$ after that. The action-value function $Q_\pi(S_t, A_t)$ can be expressed recursively as [13],

$$Q_\pi(S_t, A_t) = \mathbb{E}_\pi \left[r_t + \gamma Q_\pi(s_{t+1}, a_{t+1}) | S_t = s_t, A_t = a_t \right]. \tag{2.15}$$

Bellman's principle of optimality states that the optimal action-value function for an MDP has the recursive expression

$$Q^*_\pi(S_t, A_t) = \mathbb{E}_\pi \left[r_t + \gamma \max_{a_{t+1} \in \mathcal{A}} Q^*_\pi(s_{t+1}, a_{t+1}) | S_t = s_t, A_t = a_t \right], \tag{2.16}$$

which solution can be obtained by using a Temporal Difference (TD) algorithm [79], which solves the following update rule iteratively.

$$\hat{Q}(S_t, A_t) \doteq \hat{Q}(S_t, A_t) + \alpha \left[r_t + \gamma \max_{a_{t+1} \in \mathcal{A}} \hat{Q}(s_{t+1}, a_{t+1}) - \hat{Q}(S_t, A_t) \right], \tag{2.17}$$

in which $\hat{Q}(\cdot)$ corresponds to a function approximator used to represent $Q^*_\pi(\cdot)$ and $\alpha \in (0, 1]$ is a learning rate. Once a good quality representation of $Q^*_\pi(\cdot)$ is obtained via $\hat{Q}(\cdot)$, at time step t and state s_t, optimal actions a_t can be sampled from the optimal policy, i.e., $a_t \sim \pi^*(s_t)$, obtained as

$$\pi^*(S_t) = \max_{a \in \mathcal{A}} \hat{Q}(S_t = s_t, a). \tag{2.18}$$

For continuous state and action spaces, the optimal action-value function $Q^*_\pi(\cdot)$ can be approximated using a DNN i.e., $\hat{Q}(\cdot) = Q_\theta(\cdot)$ with parameters θ, leading to an algorithm known as deep Q-networks (DQNs) [16]. In this case, the iterative procedure shown in (2.17) can be seen as a regression problem whose objective is to estimate the DNN's parameters θ via stochastic gradient ascent. In DQNs, the Q_θ is updated using the value $r_t + \gamma \max_{a \in \mathcal{A}} Q_{\theta^{target}}(s_t, a)$, where $Q_{\theta^{\text{target}}}$ is a *target* Q-function[1]. Under this value definition, parameters θ can be obtained minimizing a loss function over mini-batches B of past data $\{(s_t, a_t, r_t, s_{t+1})\}_{i=1}^{|B|}$. In this case, the loss definition used to train the DQN is based on the mean squared Bellman error, defined as[2]

$$\min_\theta \sum_{i=1}^{|B|} \left(r_{t,i} + \gamma Q_{\theta^{\text{target}}} \left(s_{t+1,i}, \arg\max_a Q_\theta \left(s_{t+1,i}, a \right) \right) - Q_\theta \left(s_{t+1,i}, a_{t,i} \right) \right)^2. \tag{2.19}$$

Notice that in continuous action spaces, the procedure used in (2.18) to sample actions from the action-value function Q_θ is not feasible since an exhaustive action enumeration (i.e., the Max-Q problem) is not possible. Moreover, in (2.18) actions constraints are completely disregarded. To overcome this, we combine value-based DRL algorithms with mixed-integer programming, as explained next.

2.4. PROPOSED MIP-DQN ALGORITHM

The proposed DRL algorithm is named MIP-DQN and is defined through two main procedures: training and deployment (or online execution). The main objective of the training procedure is to estimate the parameters θ of the DNN used to approximate the action-value function Q_θ; whereas during deployment, the obtained function Q_θ is used to take actions to directly operate assets within the energy system. Both procedures are explained in detail below.

[1] i.e., a copy of model Q_θ which parameters are updated less frequently. This procedure helps to stabilize learning within the DRL algorithm. For a more detailed explanation, see [80].

[2] For a more detailed derivation of the loss function in (2.19), see [80].

Algorithm 1: Training procedure for MIP-DQN

Define the maximum training epochs T, episode length L. Initialize parameters of functions Q_θ, $Q_{\theta^{\text{target}}}$, and π_ω; Initialize reply buffer R. ;
for $t = 1$ **to** T **do**
  Sample an initial state s_0 from the initial distribution
  for $l = 1$ **to** L **do**
    Sample an action with exploration noise $a_t \sim \pi_\omega(s_t) + \epsilon$, $\epsilon \sim \mathcal{N}(0, \sigma)$ and observe reward r_t and new state s_{t+1}. ;
    Store transition tuple (s_t, a_t, r_t, s_{t+1}) in R.;
  Sample a random mini-batch of $|B|$ transitions (s_t, a_t, r_t, s_{t+1}) from R.;
  Update the Q-function parameters by using (2.19).;
  Update the execution policy function parameters by using $\omega \leftarrow \omega + \nabla_\omega \frac{1}{|B|} \sum_{s \in B} Q_\theta(s, \pi_\omega(s))$.
  Update the target-Q function parameters:

$$\theta^{\text{target}} \leftarrow \tau\theta + (1-\tau)\theta^{\text{target}}$$

2.4.1. Training Procedure

The training process developed for the MIP-DQN algorithm is described in Algorithm 1. This process starts by randomly initializing the parameters of the DNN functions Q_θ, $Q_{\theta^{\text{target}}}$. Then, interactions with a model of the energy system take place. In traditional valued-based RL algorithms, exploration is done by sampling actions from the current estimate of the action-value function Q_θ. However, and as explained before, sampling actions from Q_θ following (2.18) is not a feasible procedure in continuous action spaces. Instead, we propose to use a parameterized deterministic optimal policy π_ω, which is approximated using a DNN model and randomly initialized. Similar to other work [80], the policy function π_ω, the action-value functions Q_θ and $Q_{\theta^{\text{target}}}$, will be jointly approximated.

Within one epoch, for each time step t, a transition tuple of the form (s_t, a_t, r_t, s_{t+1}) is collected and store in a replay buffer R. Then, a subset B of these samples is selected and used to update the parameters of functions Q_θ, $Q_{\theta^{\text{target}}}$ and π_ω as shown in Algorithm 1. This procedure is iteratively done until a maximum number of epochs is reached.

Different from other DRL algorithms, such as DDPG and PPO, after training, we make use of the action-value function Q_θ and discard the approximated policy π_ω. Moreover, it is critical to notice that the power balance constraint is only enforced via the penalty added to the reward function in (2.13). Thus, it is expected that at the end of the training procedure, such equality constraint is not *strictly* met. The procedure used to enforce constraints is developed for the deployment or online execution, as explained next.

2.4.2. Deployment (Online Execution) Procedure

After convergence of the training procedure, the action-value function Q_θ, with fixed parameters θ, can be used to take actions to control different energy resources. To do this, the problem stated in (2.18) must be solved. In this case, as function Q_θ represents

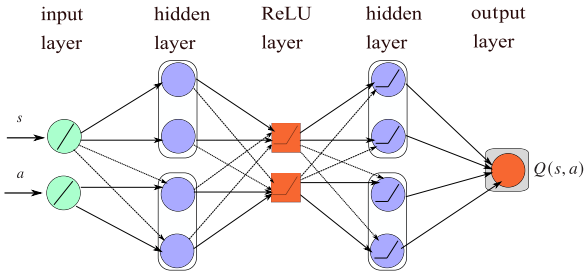


Figure 2.2: Layer structure of the DNN used to approximate the action-value function $Q(s,a)$. We denoted this DNN model as $Q_\theta(s,a)$ in Algorithm 1.

a DNN, in order to solve (2.18), we leverage recent optimization results for DNNs. Thus, proposing a transformation of the DNN model Q_θ into a MIP formulation.

MIP FOR DEEP NEURAL NETWORKS

Let the DNN $Q_\theta(s,a)$ in Fig. 2.2 consists of $K+1$ layers, listed from 0 to K. Layer 0 is the input of the DNN, while the last layer, K refers to the outputs of the DNN. Each layer $k \in \{0,1,\ldots,K\}$ have U_k units, which is denoted by $u_{j,k}$, the j_{th} unit of the layer k. Let x^k refers to the output vector of layer k, then x_j^k is the output of unit $u_{j,k}$, $(j = 1,2,\ldots,U_k)$. As layer 0 is the input of the DNN, then x_j^0 is j_{th} input value for the DNN. For each layer $k \leq 1$, the unit $u_{j,k}$ computes the output vector x^k below:

$$x^k = h\left(W^{k-1}x^{k-1} + b^{k-1}\right) \tag{2.20}$$

where W^{k-1} and b^{k-1} are matrices of weights and biases that compose the set of parameters $\theta = \{W, b\}$ and $h(\cdot)$ is the activation function, which in this case corresponds to the Rectified Linear Unit (ReLU) function, described as: for a real vector y, $\text{ReLU}(y) := \max\{0, y\}$.

Based on the above definitions, the DNN of Fig. 2.2, with fixed parameters θ, can be modeled as a valid MIP problem by modeling the ReLU function using binary constraints. Thus, using a binary activation variable z_j^k for each unit $u_{j,k}$, the MIP formulation of a DNN can be expressed as [76]:

$$\min_{x_j^k, s_j^k, z_j^k, \forall k} \left\{ \sum_{k=0}^{K} \sum_{j=1}^{l_k} c_j^k x_j^k + \sum_{k=1}^{K} \sum_{j=1}^{l_k} d_j^k z_j^k \right\} \tag{2.21}$$

Subject to:

$$\left.\begin{aligned} \textstyle\sum_{i=1}^{l_{k-1}} w_{ij}^{k-1} x_i^{k-1} + b_j^{k-1} = x_j^k - s_j^k \\ x_j^k, s_j^k \geq 0 \\ z_j^k \in \{0,1\} \\ z_j^k = 1 \rightarrow x_j^k \leq 0 \\ z_j^k = 0 \rightarrow s_j^k \leq 0 \end{aligned}\right\} \forall k, \forall j, \tag{2.22}$$

$$lb_j^0 \leq x_j^0 \leq ub_j^0, \quad j \in l_0, \tag{2.23}$$

$$\left.\begin{array}{l} lb_j^k \le x_j^k \le ub_j^k \\ \overline{lb}_j^k \le s_j^k \le \overline{ub}_j^k \end{array}\right\} \forall k, \forall j. \tag{2.24}$$

In the above formulation, weights $w_{i,j}^{k-1}$ and biases b_j^k are fixed (constant) parameters; while the same holds for the objective function costs c_j^k and d_j^k. The ReLU function output for each unit is defined by (2.22), while (2.23) and (2.24) define lower and upper bounds for the x and s variables: for the input layer ($k = 0$), these bounds have physical meaning (same limits of the Q_θ inputs i.e., s and a), while for $k \ge 1$, these bounds can be defined based on the fixed parameters θ [81]. Finally, notice that in order for the MIP formulation to be equivalent to the DNN, ReLU activation functions must be used, as explained in [76].

Enforcing Constraints in Online Execution

For an arbitrary state s_t, the optimal action a_t can be obtained by solving the MIP in (2.21)–(2.24) derived from Q_θ. In this case, as the decision variables are the actions a_t (see (2.18)), the power balance constraint in (2.4) as well as the ramp-up and ramp-down constraints in (2.6) and (2.7), respectively; can also be added to the MIP formulation described by (2.21)–(2.24). As a result, the optimal actions obtained by solving this MIP *strictly* enforce all operational constraints in the action space. This problem can be represented as,

$$\begin{aligned} \max_{a \in \mathcal{A}, x_j^k, s_j^k, z_j^k, \forall k} \quad & \{(2.21)\} \\ \text{s.t.} \quad & (2.22) - (2.24), (2.4), (2.6), (2.7). \end{aligned} \tag{2.25}$$

To better understand the MIP formulation stated in (2.25), Fig. 2.3 shows a reinterpretation of the power balance constraint in (2.4) as a hyperplane that define the feasibility region (for a three dimensional space) of the action space. Notice that such hyperplane may have different parameters for different time steps. Thus, if the hyperplane that enforces the power balance constraint is added to the MIP formulation that represents the DNN Q_θ, the solution of such mathematical problem will ensure minimum operating cost (via the maximization of Q_θ) and enforce all action space constraints, as exemplified in Fig. 2.4. In this case, this re-interpretation of the DNN as a MIP formulation offers enough flexibility to enforce equality constraints (as well as other constraints over the action space) for the energy system scheduling problem, such as the power balance. Algorithm 2 shows the step-by-step procedure used during the online execution of the proposed MIP-DQN algorithm.

2.5. Simulation Results and Discussions

In this section, simulation results and discussions are presented. A comparison with DRL algorithms available in the literature, including PPO, SAC, DDPG and TD3 algorithms, is also presented.

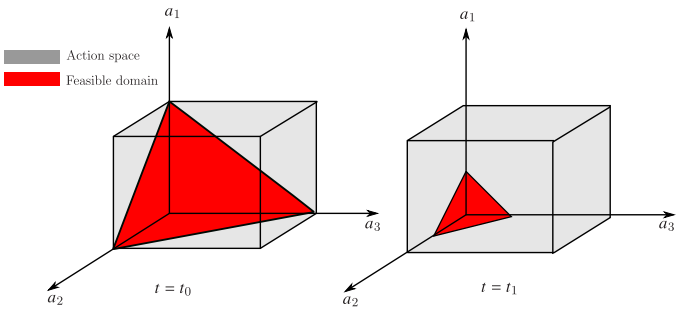


Figure 2.3: Action space (grey) and feasible action space (red) illustration. Actions a_1, a_2, a_3 refer to generic actions in a three-dimensional action space $\mathscr{A}$. For each time step t, the power balance constraint in (2.4) can be seen as the hyperplane $a_1 + a_2 + a_3 = d$ that defines the feasible actions space.

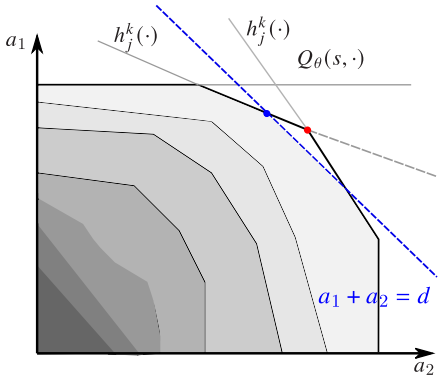


Figure 2.4: Visualization of the constraint space whose boundaries are formed by the hyperplanes $h_j^k(\cdot)$ defined by the ReLU activation functions derived from the deconstructed DNN $Q_\theta(s,\cdot)$ as a MIP formulation, for a specific state s and actions a_1 and a_2. The grey are shows the increasing value (from darker to lighter) of ∇Q_θ. The red point exemplifies the optimal solution of $\max_{a\in\mathscr{A}} Q_\theta(s,\cdot)$ if constraint $a_1 + a_2 = d$ is disregarded. If such a constraint is added to the MIP formulation, the solution represented with the blue point will be reached.

2.5.1. CASE STUDY AND SIMULATIONS SETUP

To test the developed MIP-DQN algorithm, an energy system consisting of three DG units and an ESS is defined. The DG unit's parameters are shown in Table 2.2, while for the ESS, the charging/discharging limits, nominal capacity, and energy efficiency (η_B) are set to 100 kW, 500 kW, and 0.90, respectively. We assume that the network's maximum export/import limit is defined as 30 kW. To encourage the use of renewable energies, we set selling prices as half of the current electricity prices, i.e., $\beta = 0.5$.

One-year demand consumption and PV generation data are used as the original dataset, sampled in hour resolution. Fig. 2.5 shows the mean and standard deviation of the demand consumption and PV generation during summer and winter for a period of 24h, defined as the length of one episode ($T = 24$). The original dataset is divided into two additional datasets: training and testing. The training dataset contains the first three weeks of each month, while the testing dataset contains the remaining data. This allows the DRL algorithm to learn any seasonal and weekly behavior available in the PV gen-

Algorithm 2: Online Execution for the MIP-DQN Algorithm

Extract trained parameters θ from Q_θ;
Formulate the Q-function network Q_θ as a MIP formulation according to (2.21)-(2.24). Add all action space constraints i.e., (2.4), (2.6) and (2.7).
Extract initial state s_0 based on real-time data;
for $t = 1$ **to** T **do**
  For state s_t, get optimal action by solving (2.25) using commercial MIP solvers;

Table 2.2: DG units information

Units	a[\$/kW^2]	b[\$/kW]	c[\$]	$\underline{P}^G$[kW]	$\overline{P}^G$[kW]	RU[kW]	RD[kW]
DG_1	0.0034	3	30	10	150	100	100
DG_2	0.001	10	40	50	375	100	100
DG_3	0.001	15	70	100	500	200	200

eration and demand consumption data [30]. During training, the EES's initial SOC was randomly set. To implement our MIP-DQN algorithm, PyTorch and the OMLT (see [81]) have been used. Default settings were used for all the implemented DRL algorithms, as shown in Table 2.3. All implemented algorithms are openly available in [82]. Hyper-parameters σ_1 and σ_2 are defined as 0.01 and 20, respectively, as default values. Each test is run with five random seeds to eliminate randomness from code implementation.

Table 2.3: Parameters for DRL algorithms

Algorithm	Batch size $\|B\|$	Learning rate	Buffer size R	γ	Network dimension	Optimizer
DDPG	256	1e-4	5e4	0.995	(64,64,64)	Adam
SAC	256	1e-4	5e4	0.995	(64,64,64)	Adam
TD3	256	1e-4	5e4	0.995	(64,64,64)	Adam
PPO	256	1e-4	-	0.995	(64,64,64)	Adam
MIP-DQN	256	1e-4	5e4	0.995	(64,64,64)	Adam

2.5.2. VALIDATION AND ALGORITHMS FOR COMPARISON

In the research literature, DRL algorithms are usually compared with simple rule-based or MPC-based algorithms (considering the impacts of any forecasting error) [83]. Nevertheless, this procedure does not allow us to estimate the optimality gap between current DRL algorithms and the optimal global solution with a perfect forecast of the stochastic variables (i.e., generation and demand consumption). In this case, this optimal global solution with full knowledge should be regarded as an upper boundary, as none algorithm would perform better. Based on this, to validate and fairly compare the performance of the proposed MIP-DQN algorithm, besides comparing the optimal DERs schedule defined by several state-of-the-art DRL algorithms (DDPG, PPO, TD3), we compared with the optimal solution obtained considering perfect forecast for the next 24 hours. In this case, the optimal solution is found by solving the nonlinear mathematical program-

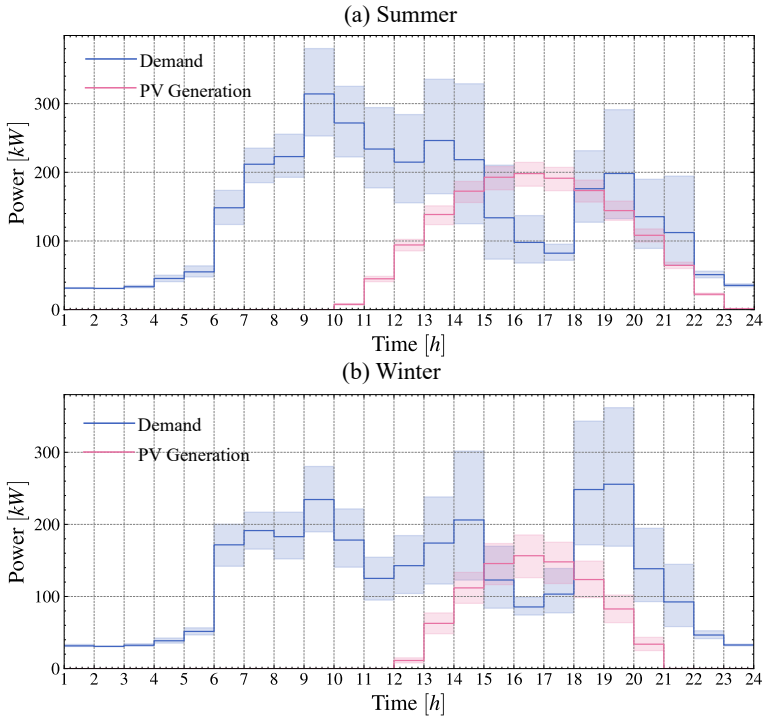


Figure 2.5: Mean and standard deviation of the demand consumption and PV generation.

ming formulation in Sec. 2.2, implemented using Pyomo [84]. Notice that different from the optimal global solution, all the tested DRL algorithms can make decisions only using current information. Finally, to evaluate the DRL algorithms' performance, the total operating cost, as in (2.1), and the power unbalance, as in (2.14), are used as metrics.

2.5.3. PERFORMANCE ON THE TRAINING SET

Figure 2.6 shows the average reward, operating cost, and power unbalance for the developed MIP-DQN algorithm and other DRL algorithms during the training process. As can be seen in Figure 2.6, the average reward increases rapidly after 100 episodes of training, while the operating cost and the power unbalance significantly decrease. This behavior during training is typical of DRL algorithms as the DNN's parameters are randomly initialized, leading initially to random actions causing high power unbalance. Throughout the training, and due to the introduction of the penalty terms used in the reward definition in (2.13), the DNN's parameters are updated, leading to higher quality actions, reducing power unbalance, and showing a lower operating cost. All algorithms converged before 400 episodes. After the last training episode, the power unbalance (presented by the average with 95% confident interval) of DDPG, SAC, PPO, and TD3 are 64.8 ± 99 kW, 807 ± 121 kW, 65 ± 18 kW, 304 ± 104 kW, respectively; while a power unbalance of 12 ± 15 kW was observed for the proposed MIP-DQN algorithm. This result shows how the proposed MIP-DQN algorithm outperformed other DRL algorithms during the train-

ing process. Nevertheless, and as expected, none of the tested DRL algorithms (including the proposed MIP-DQN) can strictly enforce the power balance; if such algorithms are used in real-time operation, they might lead to unfeasible operation. Next, we show how our proposed algorithm can overcome this during online execution, even in unseen data.

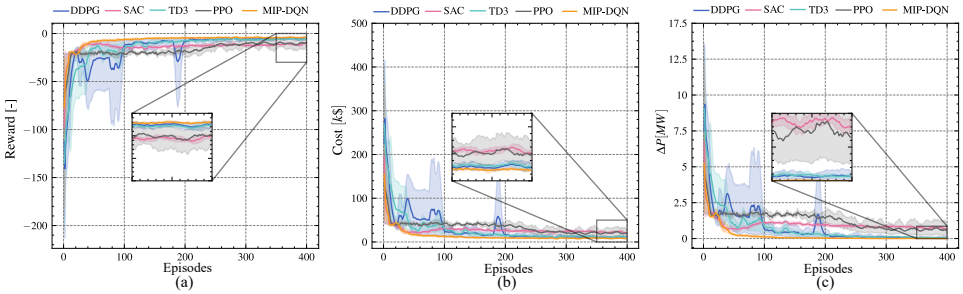


Figure 2.6: Mean and 95% confident interval for the reward, operating cost and power unbalance for the developed MIP-DQN algorithm, as well as for other DRL algorithms, during training. As expected, none of these DRL algorithms can enforce the power balance constraint.

2.5.4. PERFORMANCE ON THE TEST SET

After training, the DNN's parameters of all the DRL algorithms are fixed as shown in Algorithm 2. A performance comparison is now made on the test set. Recall that the data on the test set is not used during training; therefore, it has not been seen by any of the DRL algorithms. To compare results on the test set, Fig. 2.7 shows the cumulative operating cost and power unbalance (which can be seen as a cumulative error) for 10 different days using the proposed MIP-DQN algorithm, as well as other DRL algorithms. The optimal global solution obtained by solving the NLP formulation and considering the perfect forecast is also presented. As can be seen in Fig. 2.7, during online operation and for all 10 test days, the proposed MIP-DQN algorithm strictly meets the power balance constraint, while other DRL algorithms fail to deal with such equality constraint. Notice in Fig. 2.7 how DRL algorithms such as DDPG and TD3 reach a cumulative power unbalance near 0.14 MW at the end of the test period. As a result of such high unbalances, an operating cost of 53.3% higher than the optimal global solution is also observed. In contrast, the proposed MIP-DQN algorithm achieves an operating cost of 94 k\$, i.e., 17.6% higher than the optimal solution.

To test the performance with a higher number of test days, Table 2.4 presents the average cumulative error (with respect to the solution obtained by solving the NLP formulation with perfect forecast), the average power unbalances, and total average computational time (over 30 test days) of the proposed MIP-DQN algorithm as well as other DRL algorithms. As can be seen, the proposed MIP-DQN algorithm has the lowest average error, 13.7%; while strictly meeting the power balance (and other) constraint. In contrast, algorithms such as PPO showed poor performance reaching an error of 52.4%. As expected, the total computational time required to execute the proposed MIP-DQN algorithm is higher than other DRL algorithms. This increase in the computational time is a result of the MIP formulation required to be solved in order to enforce the equal-

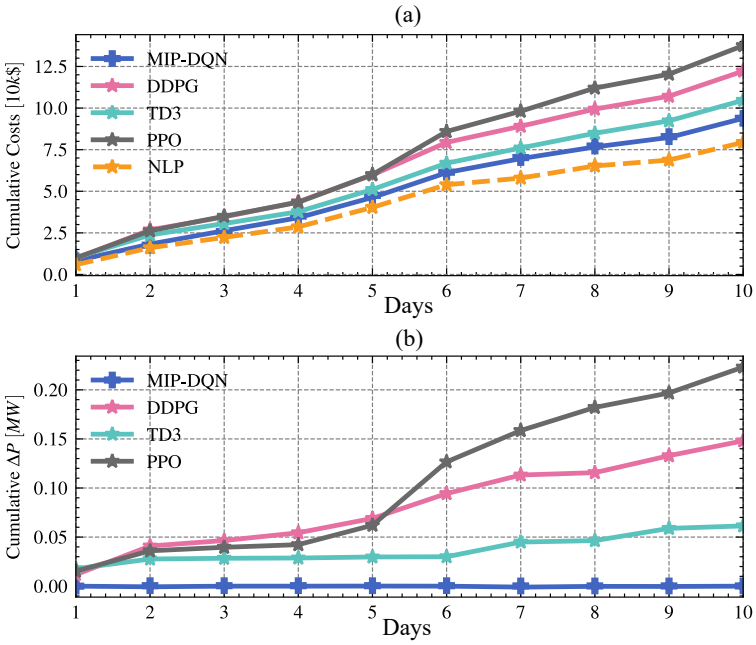


Figure 2.7: Cumulative costs and power unbalance for 10 days in the test set. The proposed MIP-DQN algorithm is able to strictly meet the power balance constraint while other DRL algorithms fail to do so.

Table 2.4: Performance comparison of different DRL algorithms in a new test set of 30 days.

Algorithms	**Error**	ΔP $[MW]$	**Computational time** $[s]$
MIP-DQN	13.7 ± 0.3%	**0.0**	17
DDPG	47.3±1.9%	0.14±0.021	4.3
TD3	31.5±0.7%	0.06±0.011	4.9
PPO	52.4±0.3%	0.15±0.007	4.3

ity constraint (see (2.25)). Nevertheless, for this case, the proposed MIP-DQN algorithm can still be used for real-time operation as it only requires less than 20 s for execution. In this case, it is important to highlight that the computation time of the proposed MIP-DQN algorithm is impacted by the size of the formulated MIP problem, which is only determined by the size of the used Q-network (layers, units of each layer, etc.) and not by the size of the energy system (microgrid) considered. Previous research has shown that (small) neural networks can generalize well in real environments [61], supporting the applicability of DRL models in real systems.

2.5.5. DISPATCH DECISIONS COMPARISON

Until now, the general performance of the proposed MIP-DQN algorithm has been presented, highlighting its capability of strictly enforcing the power balance constraint, even in unseen operational days. Next, a comparison in terms of the scheduling of the DG units and the ESSs is introduced. To do this, Fig. 2.8 displays the output power of all

the DG units, ESSs and the imported/exported power from the network for: the proposed MIP-DQN algorithm (Fig. 2.8*b*), and the optimal solution obtained after solving the NLP formulation considering perfect forecast (Fig. 2.8*c*). Notice in Fig. 2.8 that when the electricity price is high, and the net power is low, the proposed MIP-DQN algorithm dispatches the ESSs in charging mode, and a similar dispatch decision is observed in the optimal global solution. Notice also that, when compared with the optimal solution, the proposed MIP-DQN algorithm dispatched 3_{th} DG during the peak hour, which can be considered a sub-optimal decision as the operating cost of such DG is higher than the others. This difference in this dispatch decision can be due to the estimated Q-function, which might not be good enough to represent the true action-value function. In this sense, as the proposed MIP-DQN algorithm chooses actions that maximize its Q-value estimation, the largest Q-value might not represent the best action for this specific state-action pair. Nevertheless, even in executing a sub-optimal decision, the proposed MIP-DQN algorithm is able to meet the power balance constraint, guaranteeing operational feasibility. Finally, although differences in the dispatch decisions made by the proposed MIP-DQN algorithm and the optimal solution can be observed, it is important to highlight that the optimal global solution is obtained considering the perfect forecast of the future generation and demand consumption for the next 24 hours, while the proposed MIP-DQN algorithm provides dispatch decisions in an hourly basis, without knowledge of the future values of the stochastic variables.

2.5.6. Sensitivity Analysis

To better understand the impact of hyperparameter σ_2 in the reward function in (2.13), Fig. 2.9 shows the average operating cost and power unbalance (during training) for the proposed MIP-DQN algorithm for $\sigma_2 = 20, 50, 100$. As can be seen in Fig. 2.9, and as expected, higher values of σ_2 accelerate the convergence of the proposed MIP-DQN algorithm to rapidly reduce power unbalance, while having no apparent impact on the convergence of the operating cost. On the other hand, lower values of σ_2 seem to accelerate the convergence of the operating cost leaving behind the convergence of the power unbalance. In general, for the test performed, it was observed that the proposed MIP-DQN algorithm could converge in less than 200 episodes.

2.5.7. Comparison with Safe DDPG Algorithm

A comparison with current safe DRL algorithms is also performed. In this case, the proposed MIP-DQN algorithm is compared with a Safe DDPG algorithm, as presented in [85]. Fig. 2.10 shows the average reward (Fig 2.10*a*), operating cost (Fig. 2.10*b*), and power unbalance (Fig. 2.10*c*) for the two algorithms being compared. In this case, and as expected, both algorithms fail to enforce the power unbalance constraint strictly during training. At the beginning of the training stage, the Safe DDPG algorithm shows a lower operating cost and power unbalance, and higher reward, when compared to the MIP-DQN algorithm. This is mainly due to the trained linear safe layer of the Safe DDPG, which projects the exploration action to a safer one, while the MIP-DQN algorithm is free to explore the action space regardless of the feasibility of the decided action. Nevertheless, along with the training, the Safe DDPG algorithm fails to learn to reduce further or eliminate power unbalance, while our proposed MIP-DQN algorithm reduces the unbalance sharply. This behavior is mainly due to the reward shaping of the MIP-DQN

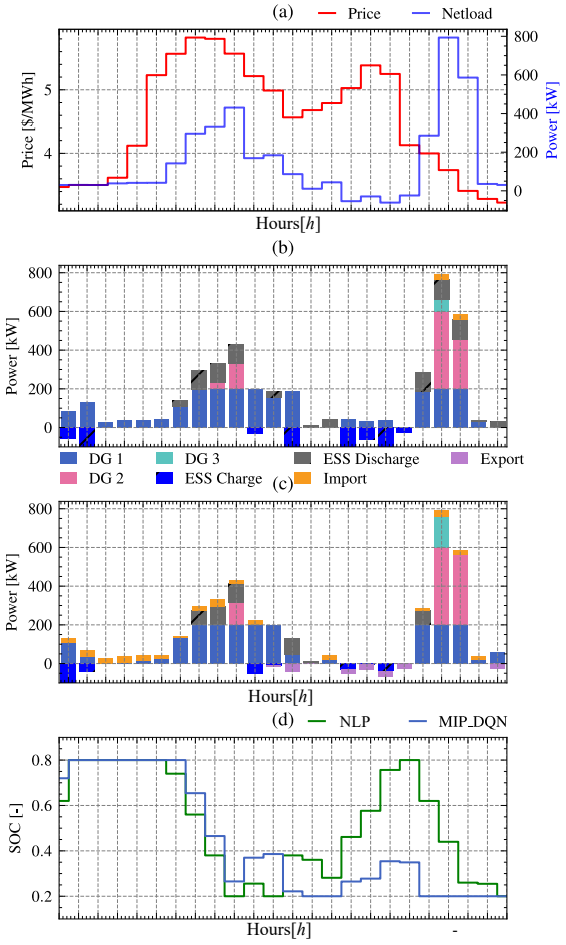


Figure 2.8: Operational schedule of all DG units and ESSs defined by the proposed MIP-DQN algorithm and the optimal global solution obtained by solving the NLP formulation considering perfect forecast.

algorithm, which can learn to avoid the penalty due to the power unbalance during the training. It is important to highlight that the performance of the Safe DDPG algorithm depends on the quality of the trained safe layer that project the original action of the DDPG algorithm to a feasible one. In this case, as the safe layer is a linear function, its generalization capabilities may not be enough to learn the complex nonlinear energy system dynamic. Thus, even after projection, the action can not fully meet the power unbalance constraint. Moreover, as the safe layer modified the action during exploration, it also harms the performance of the trained RL algorithm as shown in Fig. 2.10. Compared to the Safe DDPG algorithm, the proposed MIP-DQN algorithm learns to eliminate the unbalance in a small value after training and guarantees the feasibility during the execution (Fig. 2.7).

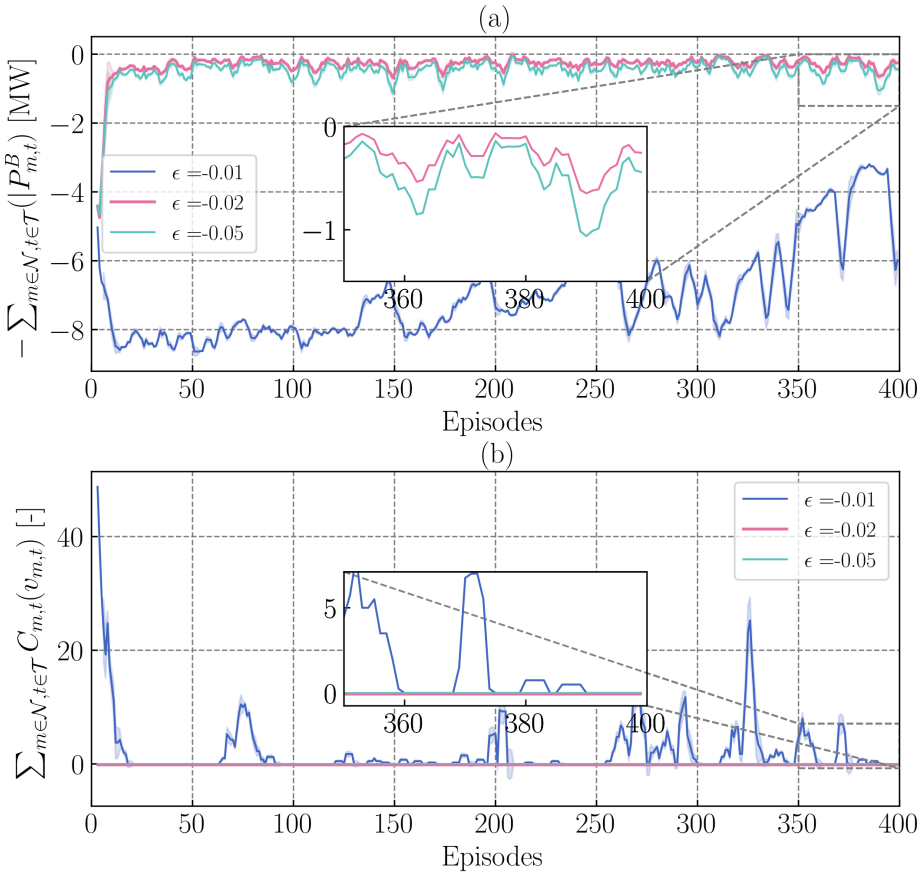


Figure 2.9: Average reward, operating cost, and power unbalance of the proposed MIP-DQN algorithm for different values of σ_2.

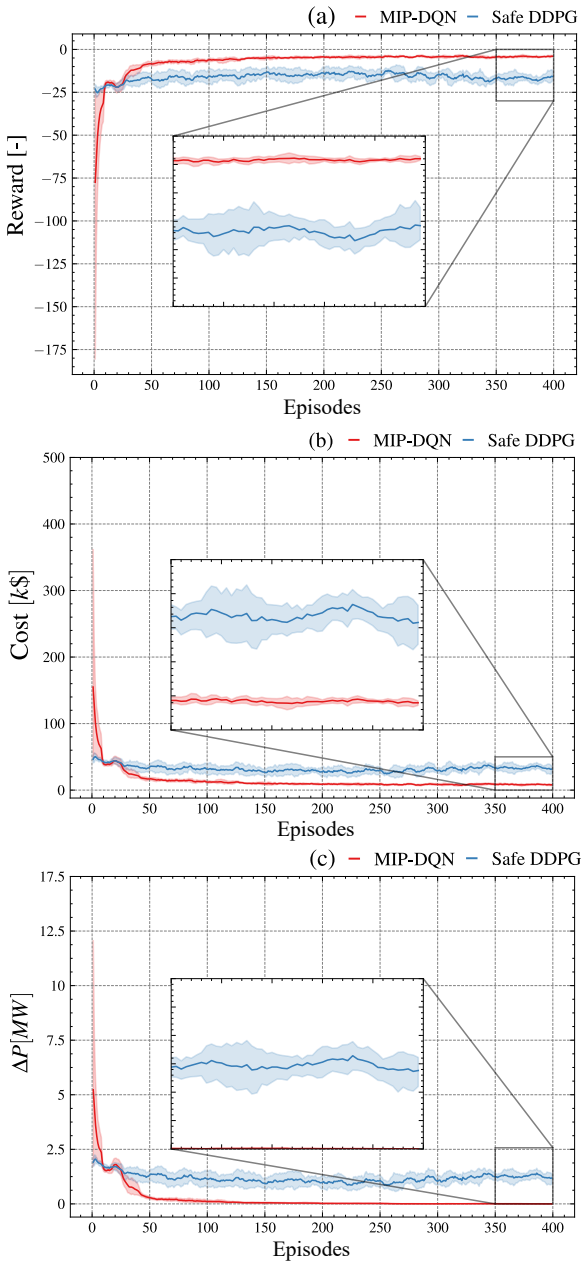


Figure 2.10: Mean and 95% confident interval for the reward, operating cost and power unbalance for the developed MIP-DQN and Safe DDPG algorithms.

2.5.8. LARGER CASE STUDY

To test the performance of the proposed MIP-DQN algorithm on an energy system with multiple ESSs, an environment with three ESSs and three DG generators is designed. For

this new environment, Fig. 2.11 shows the average operating cost and power unbalance of the proposed MIP-DQN algorithm as well as other state-of-the-art DRL algorithms, during the training process. As can be seen in Fig. 2.11, the operating cost and power unbalance are significantly reduced. In this case, all tested DRL algorithms converged at around 400 episodes. The power unbalances (presented by the average with 95% confident interval) of the DDPG, SAC, PPO and TD3 algorithms are 97 ± 125 kW, 533 ± 208 kW, 45 ± 19 kW, 462 ± 98 kW, respectively. In contrast, a power unbalance of 17 ± 22 kW was observed for the proposed MIP-DQN algorithm. Similar to the results presented in Sec. 2.5.3 for the smaller case study (see Fig. 2.6), none of the tested DRL algorithms can strictly enforce the power balance during training. Most of the observed power balance violations happen during peak load days, consistent with previous results [30]. Nevertheless, the proposed MIP-DQN algorithm is able to enforce power unbalance during the online execution, even on peak load days, as shown next. Additionally, compared to the result of simulations in Sec. 2.5.3, no performance degeneration is observed, proving the scalability of the proposed MIP-DQN algorithm.

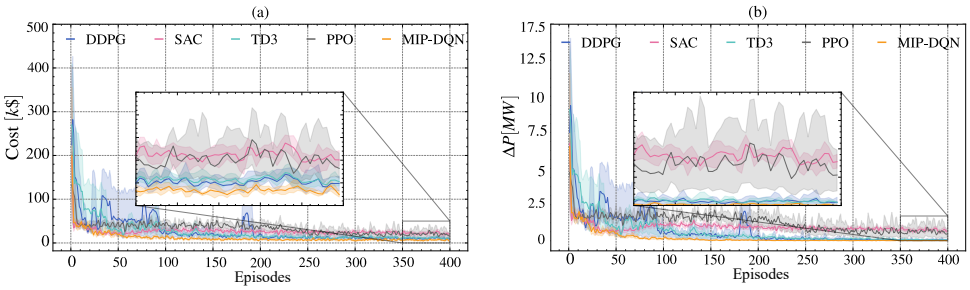


Figure 2.11: Mean and 95% confident interval for the operating cost and power unbalance for the developed MIP-DQN algorithm, as well as for other DRL algorithms, during training.

Fig.2.12 shows the scheduling decisions from the MIP-DQN algorithm for all three ESSs (Fig.2.12*b*) and DG generators (Fig.2.12*c*), and SOC changes (Fig.2.12*d*) in a typical day with extreme peak load. Notice that the power balance is strictly enforced during the peak load day. For instance, at 19h, the load is extremely high, and the MIP-DQN algorithm dispatches all the ESSs in discharging mode. This avoided importing electricity from the main grid as the electricity price was high at that particular time. These results showed that the proposed MIP-DQN algorithm learned to schedule feasible decisions for multiple ESSs in extreme peak situations. Notice also that, at hours 3 and 4, the proposed MIP-DQN algorithm dispatches the 2_{th} DG, instead of fully using the 1_{st} DG, which can be considered as a sub-optimal decision because the operating cost of 2_{th} DG is higher than that of 1_{st} DG. A similar result was observed in Fig. 2.8. Nevertheless, even in executing a sub-optimal decision, the proposed MIP-DQN algorithm is able to meet the power balance constraint, guaranteeing operational feasibility. Thus, the proposed MIP-DQN algorithm can provide feasible dispatch decisions hourly for multiple ESSs, displaying prominent scalability features.

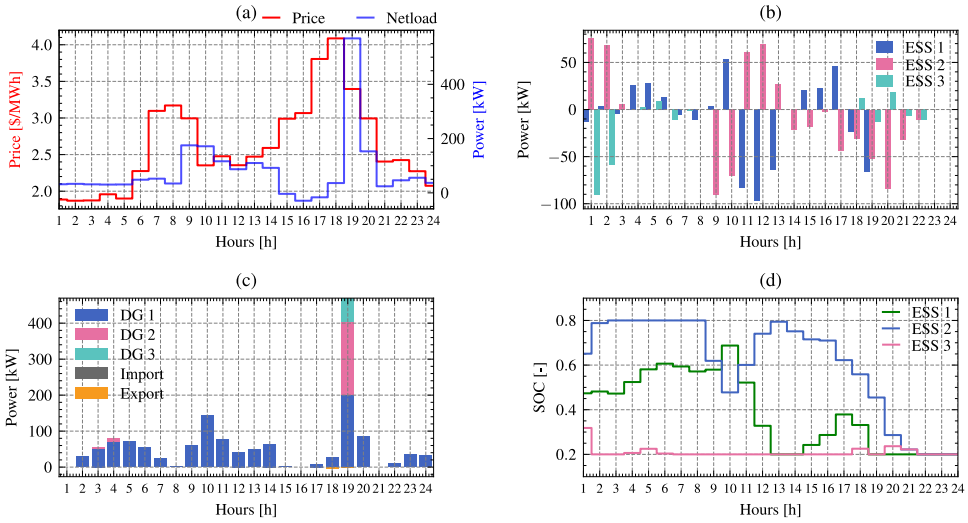


Figure 2.12: Operational schedule of all ESSs and DG units defined by the proposed MIP-DQN algorithm for a larger case study composed of three ESSs and three DG units.

2.5.9. DISCUSSION

The penetration of renewable-based DERs energies significantly increases the uncertainty and complexity of the operation of energy systems. Existing model-based approaches may not perform well when defining the operational schedule of DERs in real time due to their poor accuracy and high computational time requirements. Due to this, current efforts are put into leveraging RL algorithms' model-free and data-driven nature. After offline training, RL algorithms can provide near-optimal solutions in real-time. Nevertheless, the most critical challenge to enabling RL algorithms deployment in real energy systems scheduling frameworks is their lack of constraint enforcing guarantee. Even though several safe RL algorithms have tackled this problem, these approaches fail to meet the required security levels of energy systems operation [86]. In general, model-based optimization approaches can guarantee the feasibility of the defined DERs schedule by setting hard constraints in the mathematical formulation, which is impossible to do in current RL algorithms.

To overcome the problem mentioned above, inspired by recent advances in deep learning and optimization research areas, we first bring constraint enforcement in RL algorithms combining deep learning and optimization theory. We developed a DRL algorithm, namely MIP-DQN, that can theoretically guarantee the feasibility of the decided solution and get the optimal solution during the online scheduling stage. To do this, we redesigned the training and online-scheduling procedure. The proposed MIP-DQN algorithm uses a trained Q-network to approximate the state-action values function. Exploration and exploitation are executed based on a trained policy network to update the Q-network parameters. After training, the Q-network is assumed to approximate the optimal Q-values. Then, the trained Q-network is extracted and formulated as MIP formulation, which can be used to impose hard constraints in the action space, ensuring

the feasibility of the defined schedule. In this case, the power balance constraint is used as an example to show the effectiveness of the proposed approach. Results showed that MIP-DQN strictly meets the power balance constraint, showing a lower error when compared with other DRL algorithms and the optimal global solution.

The essence of the proposed MIP-DQN algorithm is using a trained Q-network as a surrogate function for the optimal operational decisions. As above-mentioned, the optimality is defined by the Q-network modeled as a MIP formulation. Thus, the approximation quality of the Q network determines the proposed algorithm's performance. In Fig 2.8, we showed that the proposed MIP-DQN could be considered a good quality operational schedule, albeit sub-optimal. Thus, efforts to reduce the error when compared with the optimal global solution must be centered on increasing the quality of the approximation of the Q-values via the used deep neural network. Additionally, the proposed MIP-DQN algorithm still needs to integrate a penalty term into the reward function to explore the right direction during the training process. This introduces extra hyperparameters that also impact the approximation performance of the obtained Q-function. An alternative exploration approach that can be used is to model the DNN as a MIP formulation in each iteration step; nevertheless, this would imply higher training time.

3

MIP-DRL: A Constraint Enforcement Deep Reinforcement Learning Framework for Optimal Energy Storage System Dispatch

The optimal dispatch of energy storage systems (ESSs) within distribution networks poses significant challenges, primarily due to uncertainties stemming from dynamic pricing, fluctuating demand, and the variability inherent in renewable energy sources. By exploiting the generalization capabilities of deep neural networks (DNNs), deep reinforcement learning (DRL) algorithms can learn good-quality control models that adaptively respond to distribution networks' stochastic nature. Nevertheless, the practical deployment of DRL algorithms is often hampered by their limited capacity for satisfying operational constraints in real-time, a crucial requirement for ensuring the reliability and feasibility of control actions during online operations. This paper introduces an innovative framework, named Mixed-Integer Programming Deep Reinforcement Learning (MIP-DRL), designed to overcome these limitations. The MIP-DRL framework can rigorously enforce operational constraints for optimal dispatch of ESSs throughout the online execution. The framework involves training an action-value function with DNNs, which is subsequently represented in a mixed integer programming (MIP) formulation. This

Parts of this chapter have been published in IEEE Journal of Modern Power Systems and Clean Energy with the title: *A Mix-Integer Programming Based Deep Reinforcement Learning Framework for Optimal Dispatch of Energy Storage System in Distribution Networks,* doi: 10.35833/MPCE.2024.000391. [87].

unique combination allows for the seamless integration of operational constraints into the decision-making process. We validate the effectiveness of the MIP-DRL framework through comprehensive numerical simulations, demonstrating its superior capability to enforce all operational constraints and achieve high-quality dispatch decisions. Our results show the framework's advantage over existing DRL algorithms.

3.1. INTRODUCTION

The proliferation of distributed energy resources (DERs) poses various challenges in the control and operation of electrical distribution networks [88]. Voltage issues can be seen in networks with high photovoltaic (PV) penetration and peak load scenarios. To overcome this, energy storage systems (ESSs) are increasingly being deployed, offering ancillary services to the distribution system operators (DSOs), such as voltage magnitude regulation. These ancillary services can be provided by exploiting ESSs' flexibility in response to a dynamic price throughout the day, which can be obtained by solving an optimal ESSs scheduling problem. From the ESSs operator's view, the defined ESSs dispatch should minimize operational costs while ensuring voltage magnitude constraints of the distribution network. Nevertheless, such a scheduling problem is inherently challenging due to the stochastic and uncertain nature of the dynamic prices, the demand consumption, and the renewable-based generation (e.g. from PV systems) [89].

Traditional research in optimal ESSs dispatch has predominantly focused on developing accurate models and approximated formulations that make the problem amenable for commercial solvers (e.g. [90]), collectively known as model-based approaches. Nevertheless, these model-based approaches struggle with real-time solution quality due to the increased complexity and uncertainty introduced by DERs [91]. Model-free approaches have been proposed as an alternative to overcome the shortcomings mentioned above. These approaches model the optimal ESSs scheduling problem as a Markov Decision Process (MDP) and leverage reinforcement learning (RL) algorithms to define optimal sequential decisions [92, 93]. By exploiting the good generalization capabilities of deep neural networks (DNN), DRL algorithms can perform sequential interpretations of data, learning good-quality control models that can adaptively respond to the stochastic nature of an environment [94].

Implementing DRL algorithms in a real system typically follows a two-stage process: first, an off-line initial training utilizing a simulator, and second, an online deployment of the trained algorithm into the real system [95]. This approach allows refining and rigorously testing DRL algorithms before their exposure to the realistic system. As for the optimal ESSs' dispatch problem, ensuring the feasibility and safety during the online execution of the DRL algorithms emerges as the most crucial aspect of their deployment [96]. Nevertheless, after training, standard DRL algorithms cannot provide the feasibility for defined actions during online operation, impeding the implementation of DRL algorithms in ESSs dispatch problems.

Several approaches have been developed to improve the constraint enforcement capabilities of DRL algorithms [28]. The enforcement of soft constraints is currently the most widely used approach. In this approach, a large and fixed penalty term is incorporated into the reward function when training the parameters of the control policy [97]. This allows the DRL algorithm to avoid actions that lead to unfeasible operations. Although these strategies may enforce operational restrictions during training, they cannot guarantee the feasibility of the defined operating schedule in real time, especially during peak periods of consumption and renewable generation [30]. Instead, safe DRL algorithms are implemented to directly handle constraints in distribution network operations without adding penalty terms in the reward function. In [98], a safe DRL algorithm was introduced to define a fast-charging strategy for lithium-ion batteries to enhance

the efficiency of EV charging without compromising battery safety. Utilizing the SAC-Lagrange DRL within a cyber-physical system framework; the strategy optimizes charging speeds by leveraging an electro-thermal model, outperforming existing DDPG-based and SAC-based DRL methods in terms of optimality. To ensure that the updated policy stays within a feasible set, in [99, 100], a cumulative constraint violation index is kept below a predetermined threshold. This approach was also used in [74, 75], in which the constraint violation index is designed to reflect the voltage and current magnitude violation level due to the ESSs dispatch defined. Nevertheless, enforcing constraints via cumulative indexes can only provide a probabilistic notion of safety, failing to enforce voltage and current magnitude constraints in real time due to their instantaneous nature. Alternatively, a projection operator can be developed to project actions defined by the DRL algorithm into a feasible set [101]. For instance, the projection operator proposed in [102] uses the action defined by the DRL algorithm as a starting point to solve a mathematical programming formulation, thus ensuring compliance with the constraints. A similar approach was implemented in [103] to regulate the distribution networks' voltage magnitude via the control of smart transformers. However, implementing such projection operators can degrade the performance of the DRL algorithm, as shown in [104].

A summary of the different constraint-enforcing approaches used by RL algorithms in a variety of operational problems is presented in Table 3.1. The optimal dispatch of ESSs requires meeting strict operational constraints so safety and feasibility can be guaranteed, especially during online operations. Although the safe DRL algorithms presented in Table 3.1 notably enhanced constraint enforcement capabilities and mitigated the violations significantly during the training, a significant challenge persists: these algorithms cannot provide control decisions with a theoretical guarantee of constraint enforcement in the online execution phase. This limitation poses a substantial barrier to the widespread implementation of DRL algorithms for optimal ESSs dispatch. Ensuring action feasibility in real-time applications is paramount, not only for the reliability of the ESSs operation but also for the broader adoption and trust in DRL solutions within this field. In our previous work [105], a value-based safe DRL algorithm is proposed to address the energy management problem with strict enforcement of the constraint of power balance equality. The work integrates optimization techniques with DRL theory, representing train Q-function networks in deep Q-learning as a MIP formulation. Leveraging this innovative approach, we now broaden the scope of our research to conceptualize and develop a more versatile and comprehensive framework that strictly enables state-of-the-art (SOTA) actor-critic DRL algorithms to enforce operational constraints. This framework is called Mixed-Integer Programming Deep Reinforcement Learning (MIP-DRL). Distinct from our earlier contribution, MIP-DRL is not confined to a specific algorithm but is envisioned as a general framework that can empower many standard actor-critic DRL algorithms to enforce operational constraints. Our contributions are systematically structured to highlight the innovation and applicability of the MIP-DRL framework, as follows:

- We present the MIP-DRL framework to enforce operational constraints with strict adherence during online operations. Utilizing the robust constraint-enforcing ability of MIP, the framework ensures compliance with operational constraints, guaranteeing zero-constraint violations during the online execution phase. This inno-

Table 3.1: Summary of literature in safe DRL algorithms and constraint enforcing approaches in energy systems operation.

Work	Research Problem	Constraint Enforcing	Open-access
[106]	Microgrid operation	Penalty function	No
[26]	Voltage regulation		Yes
[50]	Optimal power flow		No
[54]	Energy dispatch		No
[57]	Energy dispatch		No
[30]	Optimal energy system scheduling		Yes
[107]	Home energy management	Primal-dual DDPG	No
[108]	EV in Microgrid	Primal-dual SAC	No
[109]	Microgrid energy management	Constrained policy optimization	No
[110]	Cooling system control	Gaussian process	No
[98]	EV charge/discharge operation	Lagrange-SAC	No
[111]	Distribution network operation	Safe layer	No
[112]	Voltage regulation	Safe layer	Yes
[105]	Micro-grid operation	Q-network formulated MIP	Yes
[70]	Energy management	Safe layer	No
[72]	Energy hub trading	Gaussian process Safe layer	
[73]	Microgrid operation	Action projection	
[74]	Distribution network operation	Constrained policy optimization	No
[75]	EV management		

vation extends the theoretical underpinnings of the DRL applicability and enables the feasibility of its real-time applications.

- The MIP-DRL framework broadens its utility across diverse DRL algorithms that employ DNNs for Q-function approximation. We implemented and tested the proposed framework with SOTA standard actor-critic algorithms such as DDPG, TD3, and SAC, demonstrating the capability to enforce operational constraints strictly.
- Demonstrating its practical efficacy, the MIP-DRL framework is used to address the complex challenge of optimal ESSs dispatch problem in distribution networks. The results illustrate the performance superiority of the MIP-DRL framework over existing standard (safe) DRL algorithms to improve performance and ensure action feasibility, even in unseen scenarios (data).

3.2. MATHEMATICAL FORMULATION

The optimal scheduling of ESSs in a distribution network can be modeled using the nonlinear programming (NLP) formulation given by (3.1)–(3.11). The objective function in (3.1) aims to minimize the total operational cost over the time horizon $\mathcal{T}$, comprising the cost of importing power from the main grid. The operational cost ρ_t at time slot t is settled according to the day-ahead market prices ρ_t in EUR/kWh.

$$\min_{P^B_{m,t}, \forall m \in \mathcal{B}, \forall t \in \mathcal{T}} \left\{ \sum_{t \in \mathcal{T}} \left[\rho_t \sum_{m \in \mathcal{N}} \left(P^D_{m,t} + P^B_{m,t} - P^{PV}_{m,t} \right) \Delta t \right] \right\}. \tag{3.1}$$

Subject to:

$$\sum_{nm\in\mathscr{L}} P_{nm,t} - \sum_{mn\in\mathscr{L}} (P_{mn,t} + R_{mn} I^2_{mn,t}) + P^B_{m,t} + P^{PV}_{m,t} + P^S_{m,t} = P^D_{m,t} \quad \forall m \in \mathscr{N}, \forall t \in \mathscr{T} \tag{3.2}$$

$$\sum_{nm\in\mathscr{L}} Q_{nm,t} - \sum_{mn\in\mathscr{L}} (Q_{mn,t} + X_{mn} I^2_{mn,t}) + Q^S_{m,t} = Q^D_{m,t} \forall m \in \mathscr{N}, \forall t \in \mathscr{T} \tag{3.3}$$

$$V^2_{m,t} - V^2_{n,t} = 2(R_{mn} P_{mn,t} + X_{mn} Q_{mn,t}) + (R^2_{mn} + X^2_{mn}) I^2_{mn,t} \quad \forall m,n \in \mathscr{N}, \forall t \in \mathscr{T} \tag{3.4}$$

$$V^2_{m,t} I^2_{mn,t} = P^2_{mn,t} + Q^2_{mn,t} \quad \forall m,n \in \mathscr{N}, \forall t \in \mathscr{T} \tag{3.5}$$

$$SOC^B_{m,t} = SOC^B_{m,t-1} + \begin{cases} \frac{\eta^B_{m,c} P^B_{m,t} \Delta t}{\overline{E}^B_m}, & \text{if } P^B_{m,t} > 0 \\ \frac{P^B_{m,t} \Delta t}{\eta^B_{m,d} \overline{E}^B_m}, & \text{if } P^B_{m,t} < 0 \end{cases} \quad \forall m \in \mathscr{B}, \forall \sqcup \in \mathscr{T} \tag{3.6}$$

$$\underline{SOC}^B_m \le SOC^B_{m,t} \le \overline{SOC}^B_m \quad \forall m \in \mathscr{B}, \forall t \in \mathscr{T} \tag{3.7}$$

$$\underline{P}^B_m \le P^B_{m,t} \le \overline{P}^B_m \quad \forall m \in \mathscr{B}, \forall t \in \mathscr{T} \tag{3.8}$$

$$\underline{V}^2 \le V^2_{m,t} \le \overline{V}^2 \quad \forall m \in \mathscr{N}, \forall t \in \mathscr{T} \tag{3.9}$$

$$0 \le I^2_{mn,t} \le \overline{I}^2_{mn} \quad \forall mn \in \mathscr{L}, \forall t \in \mathscr{T} \tag{3.10}$$

$$P^S_{m,t} = Q^S_{m,t} = 0 \quad \forall m \in \mathscr{N} \setminus \{1\}, \forall t \in \mathscr{T} \tag{3.11}$$

The steady-state operation of the distribution network is modeled by the load flow sweep method as is shown in (3.2)–(3.5) in terms of the active $P_{mn,t}$ power, reactive power $Q_{mn,t}$ and current magnitude $I_{mn,t}$ of lines, and the voltage magnitude $V_{m,t}$ of nodes [113]. Equation in (3.6) models the dynamics of the ESSs' SOC on the set $\mathscr{B}$, while (3.7) enforces the SOC limits. Hereafter, it is assumed that the ESS $m \in \mathscr{B}$ is connected to node m, thus, $\mathscr{B} \subseteq \mathscr{N}$. Finally, (3.8) enforces the ESSs discharge/charge operation limits, (3.9) and (3.10) enforce the voltage magnitude and line current limits, respectively, while (3.11) enforces that only one node is connected to the substation. Notice that to solve the above-presented NLP formulation, all long-term operational data (e.g., expected PV generation and consumption) must be collected to properly define the ESSs' dispatch decisions, while the power flow formulation must also be considered to enforce the voltage and current magnitude limits.

In the formulated problem, we assumed that only PV panels and ESSs are installed in the distribution networks. The active power flexibility provided by ESSs dispatches is used to provide economic benefits and ensure safe voltage magnitude levels for the distribution network. It should be mentioned that the complexity of ESSs model can be increased, including a detailed physical dynamic model e.g., efficiency curves, temperature, and degradation. However, since this work aims to assess the performance of the proposed MIP-DRL framework, then ESSs dynamics are simplified by using the linear model [59].

3.3. ESSs SCHEDULING PROBLEM MDP FORMULATION

The above-presented mathematical formulation can be modeled as a finite MDP, represented by a 5-tuple $(\mathscr{S},\mathscr{A},\mathscr{P},\mathscr{R},\gamma)$, where $\mathscr{S}$ represents a state space, $\mathscr{A}$ an action space, $\mathscr{P}$ the state transition probability function, $\mathscr{R}$ the reward function, and γ a discount factor. The decision as to which action a_t is chosen in a particular state s_t is governed by a policy $\pi(a_t|s_t)$. In a standard RL algorithm, an RL agent employs the policy $\pi(a_t|s_t)$ to interact with the formulated MDP defining a trajectory of states, actions, and rewards: $\tau = (\boldsymbol{s}_0, \boldsymbol{a}_0, \boldsymbol{s}_1, \boldsymbol{a}_1, \cdots)$. Here, the RL agent's goal is to estimate a policy that maximizes the expected discounted return $J(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\mathscr{T}} \gamma^t r_t\right]$, in which $\mathscr{T}$ is the length of the control horizon.

Different from the standard RL approach, in a constrained MDP, the RL agent aims to estimate a policy π confined in a feasible set $\Pi_C = \{\pi : J_{C_i}(\pi) \le 0, \quad i = 1,\ldots,k\}$. Here, $J_{C_i}(\pi)$ denotes a cost-based constraint function induced by the constraint violation functions $C_{i,t}(\cdot)$, $i = 1,\cdots,k$. Based on these definitions, a constrained MDP can be formulated as the constrained optimization problem:

$$\begin{aligned} &\max_{\pi} J(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\mathscr{T}} \gamma^t r_t\right] \\ &\text{s.t. } J_{C_i}(\pi) \le 0, \forall i = 1,\ldots,k. \end{aligned} \tag{3.12}$$

Here, $J_{C_i}(\pi)$ is defined as $J_{C_i}(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\mathscr{T}} \gamma^t C_{i,t}\right]$. A more detailed MDP description of the ESSs optimal scheduling problem is presented below.

The state s_t denotes the operating status information of the network, which the agent can observe. The state at t is defined by $s_t = [P^N_{m,t}|_{m\in\mathscr{N}}, \rho_t, SOC^B_{m,t}|_{m\in\mathscr{B}}, t]$, where $P^N_{m,t} = P^D_{m,t} - P^{PV}_{m,t}$ corresponds to the nodal net power. These features can be divided into endogenous and exogenous features. The former includes the PV generation $P^{PV}_{m,t}$ and consumption $P^D_{m,t}$, day-ahead price ρ_t and current time step t, which are independent of the operated actions, while the latter includes the ESSs' SOC $SOC^B_{m,t}$, which depends on the agent's action and previous state s_{t-1}.

The action at time t is defined as $a_t = [P^B_{m,t}|_{m\in\mathscr{B}}]$, in which $a_t \in \mathscr{A}$, while $\mathscr{A}$ is a continuous space. Notice that a_t refers to the charging/discharging dispatch for the m_{th} ESS connected to node m in the distribution network.

Given the state s_t and action a_t at time step t, the system transit to the next state s_{t+1} defined by the next transition probability

$$p(S_{t+1}, R_t|S_t, A_t) = \Pr\{S_{t+1} = s_{t+1}, R_t = r_t \mid S_t = s_t, A_t = a_t\}. \tag{3.13}$$

This transition probability function $\mathscr{P}$ models the endogenous distribution network and ESSs dynamics, determined by the physical model of the electrical network and ESSs, and the exogenous uncertainty caused by the PV generation, demand consumption, and the day-ahead price dynamics. In practice, building an accurate mathematical model for such a transition function is not possible. Nevertheless, model-free RL algorithms do not require prior knowledge of function $\mathscr{P}$ as it can be implicitly learned by interacting with the environment.

RL algorithms can learn representative operation strategies from interactions with the environment. To achieve this goal, the environment must provide a reward r_t to

quantify the goodness of any action taken during the interaction process. In this case, the raw reward is defined as the negative of the operational cost for the operation of the distribution network, modeled defined as:

$$\mathscr{R}_t(s_t, a_t) = r_t = -\rho_t \left[\sum_{m \in \mathscr{N}} \left(P^D_{m,t} + P^B_{m,t} - P^{PV}_{m,t} \right) \right] \Delta t \tag{3.14}$$

3.3.1. OPERATIONAL CONSTRAINTS

DRL algorithms optimize operational costs while adhering to the operational constraints of ESSs and the distribution network. These constraints include the ESSs' maximum discharge/charge capacities (3.8), maintaining the SOC within specified limits (3.7), and ensuring voltage magnitude (3.9) remain within boundaries. While constraints on action spaces (3.7 and 3.8) are straightforward to enforce through action boundaries, voltage magnitude current constraints (3.9) require addressing the physical dynamics of the distribution network. To manage these limits, constraint violation functions $C_{m,t}$ are integrated into the reward function (3.14) as penalties, converting the constrained optimization problem (3.12) into an unconstrained one, redefined as:

$$r_t = -\rho_t \left[\sum_{m \in \mathscr{N}} \left(P^D_{m,t} + P^B_{m,t} - P^{PV}_{m,t} \right) \right] \Delta t - \sigma \left[\sum_{m \in \mathscr{B}} C_{m,t}(V_{m,t}) \right], \tag{3.15}$$

where σ balances operational costs against penalties for constraint violations. The constraint violation function $C_{m,t}$ in (3.15) can be modeled using different functions (e.g., L_2 functions). Here, as in [59], $C_{m,t}$ is defined as

$$C_{m,t} = \min \left\{ 0, \left(\frac{\overline{V} - \underline{V}}{2} - \left| V_0 - V_{m,t} \right| \right) \right\}, \forall m \in \mathscr{B}. \tag{3.16}$$

Nevertheless, it is critical to notice that enforcing operational constraints by only adding a penalty term into the reward function during the training can lead to infeasible operational states during the online execution, as observed in [30]. To address this, we propose the MIP-DRL framework, leveraging constraint-enforcing capabilities of MIP to ensure feasible solutions during online execution.

3.4. CONSTRAINT ENFORCEMENT MIP-DRL FRAMEWORK

The proposed MIP-DRL framework is defined through two main procedures: *(i) Training*, where the action-value function is approximated, and *(ii) Deployment*, executed during online decision-making. Both of these procedures are explained in detail below.

3.4.1. STEP-BY-STEP TRAINING

The step-by-step training for the MIP-DRL framework integrates concepts from actor-critic DRL algorithms—DDPG [23], TD3 [24], and SAC [77]—within a unified training procedure. Fig. 3.1 illustrates the interaction of the actor $\pi_\omega(\cdot)$ (also known as policy) and critic $Q_\theta(\cdot)$ (also known as action-value function) models with the environment (distribution network). Initially, the actor $\pi_\omega(\cdot)$ and critic $Q_\theta(\cdot)$ models' parameters are ran-

domly initialized. Training progresses through interaction with the environment: actions a_t are sampled from the actor model, prompting the environment to transition to new states and generate rewards (Fig. 3.1 (a)). These state transitions and rewards inform the storage of transition tuples (s_t, a_t, r_t, s_{t+1}) in a replay buffer R. Subsequently, subsets of these tuples are used to iteratively update the actor and critic models, enhancing the policy's performance and the accuracy of the action-value function estimation (Fig. 3.1 (b)).

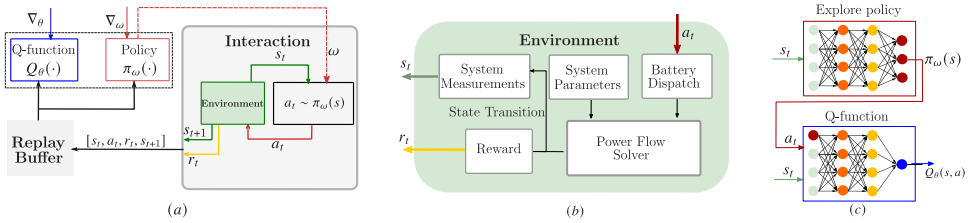


Figure 3.1: Training procedure of the proposed MIP-DRL framework. The training procedure of the MIP-DRL follows actor-critic approaches, while detail network structure can be determined by the algorithm used in the framework: (*a*) Interaction with the environment is done by sampling actions from a policy model $\pi_\omega(s)$. Information collected from the environment in the form of tuples (s_t, a_t, r_t, s_{t+1}) are stored in a replay buffer R and later used to update the parameters of the policy $\pi_\omega(s)$ (or actor) and action-value function $Q_\theta(s,a)$ (or critic) model. (*b*) Environment (distribution network) composed of a power flow solver. After implementing the current actions a_t (ESSs dispatch schedule), the environment provides the reward r_t as in (3.15) and the state transition to define s_{t+1} via (simulated) network measurements. (*c*) Policy model $\pi_\omega(s)$, which defines the action a_t, and the action-value function $Q_\theta(s,a)$, which assess the quality of the defined action a_t for state s_t.

In general, the main objective of actor-critic algorithms is to approximate a good policy network $\pi_\omega(\cdot)$ while the action-value function is used during exploration to improve the quality of the policy network. After training, the action-value function $Q_\theta(\cdot)$ is discarded. Different from this procedure, the developed MIP-DRL framework follows the actor-expert definition [114], which aims to get an optimal action based on the optimal action-value function Q_θ. Thus, during training, the policy network $\pi_\omega(\cdot)$ is only used to explore and exploit new states and actions to improve the quality of action-value function $Q_\theta(\cdot)$, while the policy network π_ω is discarded. Once a good quality representation of $Q^*_\pi(\cdot)$ is obtained via $\hat{Q}(\cdot)$, at time step t and state s_t, optimal actions a_t can be sampled from the optimal policy, i.e., $a_t \sim \pi^*(s_t)$, obtained as

$$\pi^*(S_t) = \max_{a \in \mathcal{A}} \hat{Q}(S_t = s_t, a). \tag{3.17}$$

As a result, the training procedure developed for the MIP-DDPG, MIP-TD3, and MIP-SAC algorithms resembles the training procedure of their standard DRL algorithms counterparts. Nevertheless, actions defined using only such action function $Q_\theta(\cdot)$ cannot strictly enforce operational constraints during the online execution. To overcome this, the proposed framework leverages the MIP formulation of the trained Q-function $Q_\theta(s,a)$ to enforce operational constraints during online execution, as explained next.

3.4.2. ENFORCING CONSTRAINTS IN ONLINE EXECUTION

The trained action-value function, $Q_\theta(\cdot)$, obtained from MIP-DRL algorithms with fixed parameters θ can be transformed into a MIP model, facilitating operational constraint enforcement during online execution. This transformation enables the incorporation of system constraints directly into the action decision process, as detailed in our previous work [105].

Based on the definitions in our prior work [105], the action-value function $Q_\theta(\cdot)$ obtained from trained MIP-DRL algorithms with fixed parameters θ, can be modeled as a valid MIP problem expressed as [76],

$$\max_{x_j^k, s_j^k, z_j^k, \forall k} \left\{ \sum_{k=0}^{K} \sum_{j=1}^{l_k} c_j^k x_j^k + \sum_{k=1}^{K} \sum_{j=1}^{l_k} d_j^k z_j^k \right\} \tag{3.18}$$

Subject to:

$$\left.\begin{aligned} \textstyle\sum_{i=1}^{l_{k-1}} w_{ij}^{k-1} x_i^{k-1} + b_j^{k-1} = x_j^k - s_j^k \\ x_j^k, s_j^k \geq 0 \\ z_j^k \in \{0,1\} \\ z_j^k = 1 \rightarrow x_j^k \leq 0 \\ z_j^k = 0 \rightarrow s_j^k \leq 0 \end{aligned}\right\} \forall k, \forall j, \tag{3.19}$$

$$lb_j^0 \leq x_j^0 \leq ub_j^0, \quad j \in l_0, \tag{3.20}$$

$$\left.\begin{aligned} lb_j^k \leq x_j^k \leq ub_j^k \\ \overline{lb}_j^k \leq s_j^k \leq \overline{ub}_j^k \end{aligned}\right\} \forall k, \forall j. \tag{3.21}$$

Each layer $k \in \{0,1,\ldots,K\}$ in DNN formulated Q-funtion has U_k units, denoted by $u_{j,k}$, with j being the unit index in layer k. We denote the output vector of layer k as x^k, where x_j^k is the output of unit $u_{j,k}$, $(j = 1,2,\ldots,U_k)$. Weights $w_{i,j}^{k-1}$ and biases b_j^k are fixed (constant) parameters; the same holds for the objective function costs c_j^k and d_j^k. The activation function output for each unit is defined by (3.19), while (3.20) and (3.21) define lower and upper bounds for the x and s variables: for the input layer ($k = 0$), the inputs x^0 corresponds to the same inputs of the Q-function $Q_\theta(\cdot)$, i.e. state s_t and action a_t, while the defined bounds have physical meaning (same limits of the Q_θ inputs). For $k \geq 1$, the bounds are defined based on the fixed parameters, as explained in θ [105].

Then, the max-Q problem for Q-function Q_θ in (3.17) is equivalent to (3.18)–(3.21) formulation [76]. In this case, as the decision variables are the actions a_t (corresponding to the charging/discharging schedule of the ESSs), the voltage magnitude constraints in (3.9), as well as the ESSs SOC dynamics and the discharge/charge operation limits, in (3.7) and (3.8), respectively; can all be added on top of (3.18)–(3.21). As a result, the optimal actions obtained by solving this MIP formulation *strictly* enforce all the actions and environment's operational constraints[1]. This integrated MIP problem can be repre-

[1] A general mathematical proof of optimality for the proposed MIP-DRL framework is presented in our previous work in [105].

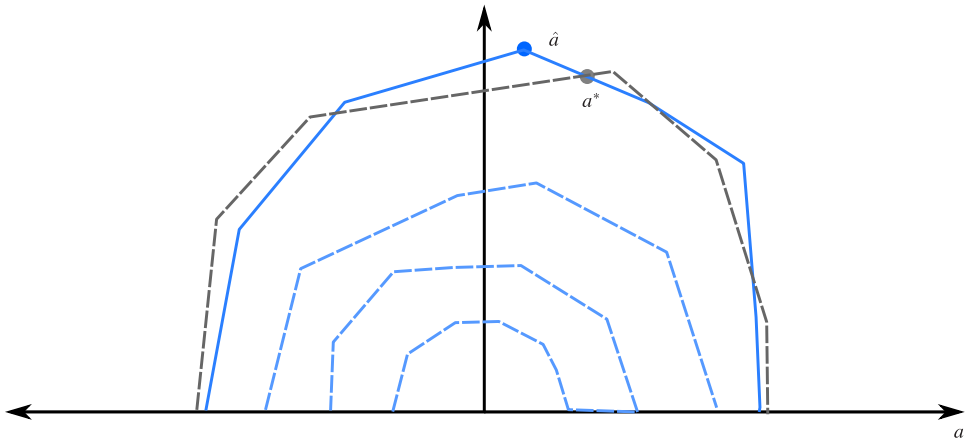


Figure 3.2: Visualization of the linear space (within the blue line) whose boundaries are formed by the hyperplanes $h_j^k(\cdot)$ defined by the Rectified Linear Unit (ReLU) activation functions derived from the deconstructed DNN $Q_\theta(s,\cdot)$ as a MIP formulation. The blue point represents the optimal solution of the problem in (3.17), which defines the optimal action $\hat{a}$. Notice also that $\hat{a}$ also corresponds to the solution of the MIP problem in (3.18)–(3.21). The linear space (within the dashed grey line) represents the hyperplanes defined by constraints (3.7), (3.8) and (3.9). Solving the MIP formulation by considering constraints (3.7), (3.8) and (3.9), solution a^* is obtained. Solution a^* corresponds to the optimal solution of (3.17) that enforces operational constraints (3.7), (3.8) and (3.9).

sented as

$$\max_{a\in\mathscr{A},x_j^k,s_j^k,z_j^k,\forall k} \left\{ \sum_{k=0}^{K}\sum_{j=1}^{l_k} c_j^k x_j^k + \sum_{k=1}^{K}\sum_{j=1}^{l_k} d_j^k z_j^k \right\} \tag{3.22}$$

$$\text{s.t.} \quad (3.19) - (3.21), (3.7), (3.8), (3.9).$$

To better understand the above-stated MIP formulation, Fig. 3.2 shows a visual representation of the MIP formulation in (3.18)–(3.21). Such formulation defines the linear space within the blue line whose boundaries are formed by the hyperplanes $h_j^k(\cdot)$, defined by the activation functions derived from the deconstructed DNN $Q_\theta(s,\cdot)$ [115]. In Fig. 3.2, the blue point represents the optimal solution of the problem in (3.17), denoted as $\hat{a}$. Notice that $\hat{a}$ also corresponds to the solution of the MIP problem in (3.18)–(3.21). Similarly, the set of constraints (3.7), (3.8) and (3.9) forms the linear space represented within the dashed grey line. Therefore, solving the MIP formulation in (3.22) provides solution a^*, which represents the optimal solution of problem (3.17) that simultaneously enforces the operational constraints defined by (3.7), (3.8) and (3.9).

The online execution phase (Algorithm 3) operationalizes this MIP representation, incorporating not only the structure of $Q_\theta(s,a)$ but also system-specific constraints (e.g., voltage magnitude constraints). By solving the MIP problem (3.22), we obtain actions a_t that maximize the expected reward while strictly adhering to operational constraints, thus ensuring the feasibility and optimality of the decisions made by the proposed MIP-DRL framework.

Algorithm 3: Online Execution for all the proposed MIP-DRL Algorithms

Extract trained parameters θ from Q_θ;
Formulate the action-value function Q_θ as a MIP formulation according to (3.18)-(3.21). Add on the operational constraints i.e., (3.7), (3.8), (3.9).
Extract initial state s_0 based on real-time data;
for $t = 1$ **to** T **do**
  For state s_t, get optimal action by solving (3.22) using commercial MIP solvers;

3.5. Simulation Results and Discussions

3.5.1. Simulation Setup

Environment Data and Framework Implementation

To demonstrate the effectiveness of the proposed MIP-DRL framework, a modified 34-node IEEE test distribution network is used, as is shown in Fig. 3.3. ESSs are placed at nodes 12, 16, 27, 30 and 34 due to their higher chance of over- and undervoltage issues. The training data used corresponds to historical Dutch market day-ahead prices, while load and PV generation measurements with a 15-minute resolution are provided by a distribution network operator. The original one year dataset is divided into two additional datasets: training and testing. The training dataset contains the first three weeks of each month, while the testing dataset contains the remaining data. This allows the DRL algorithm to learn any seasonal and weekly behavior available in the PV generation and load consumption data [59].

Table 3.2 summarizes the key parameters used for the MIP-DDPG, MIP-TD3, and MIP-SAC algorithms. This includes the discount factor, optimizer type, learning rate, batch size, and replay buffer size for each algorithm. Additionally, specific parameters for the entropy in the MIP-SAC algorithm, the reward function, and the operational limits for ESSs listed. The voltage magnitude limits are defined as $\overline{V} = 1.05$ and $\underline{V} = 0.95$ p.u. PyTorch and OMLT (see [81]) packages have been used to implement our MIP-DRL framework. Default settings shown in Table 3.2 were used for all the implemented DRL algorithms. The formulated MIP-DRL algorithms are solved with Gurobi [116].

Validation and Benchmarks for Comparison

To demonstrate the superior performance of our proposed MIP-DRL algorithms (MIP-DDPG, MIP-TD3, and MIP-SAC), we compare their scheduling outcomes with those of standard DRL algorithms (DDPG, TD3, SAC) and a safe DRL algorithm, Safe DDPG [2]. Our comparison utilizes two key metrics: operational cost in EUR, which reflects the economic efficiency of the schedules, and the number of voltage magnitude violations, indicating the algorithms' ability to enforce constraints during online execution. Furthermore, we have also compared them with the optimal global solution obtained, considering a perfect forecast for the next 24 hours. This optimal solution is obtained by solving the NLP formulation in Sec. 3.2, implemented using Pyomo and IPOPT solver.

[2] The hyperparameters for DDPG, TD3, and SAC are aligned with those of the MIP-DRL algorithms. For Safe DDPG, we adopt a linear safe layer and follow the default parameter settings as described in [85].

Table 3.2: Summary - Parameters for DRL algorithms and the MDP

MIP-DDPG Alg.	$\gamma = 0.995$ Optimizer = Adam Learning rate=$6e-4$ Batch size = 512 Replay buffer size(λ) = $4e5$
MIP-TD3 Alg.	$\gamma = 0.995$ Optimizer = Adam Learning rate=$6e-4$ Batch size = 512 Replay buffer size = $4e5$
MIP-SAC Alg.	$\gamma = 0.995$ Optimizer = Adam Learning rate=$6e-4$ Batch size = 512, Replay Buffer = $4e5$ Entropy=fixed
Reward	$\sigma = 400$
ESSs	$\overline{P}^B = 100kW, \underline{P}^B = -100kW,$ $\overline{SOC}^B = 0.8, \underline{SOC}^B = 0.2, \eta_c^B/\eta_d^B = 0.98$

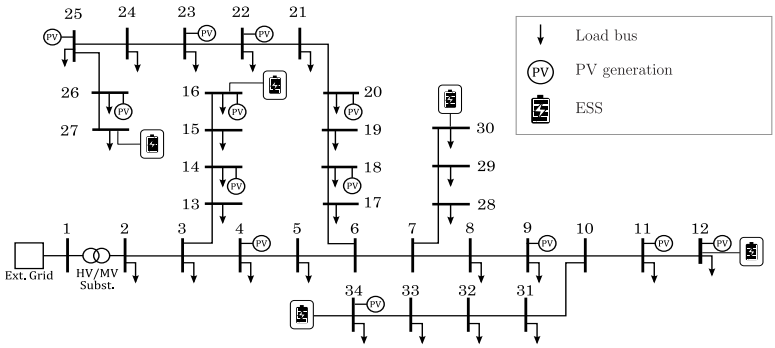


Figure 3.3: Modified IEEE-34 Node bus test system with distributed PV generation and ESSs. The ESSs are placed at the end of each feeder to increase the number of voltage magnitude issues experienced.

3.5.2. PERFORMANCE OF MIP-DRL ALGORITHMS ON THE TRAINING SET

Figure 3.4 displays the average total reward, operational cost, and the number of voltage magnitude violations during the training process for the developed MIP-DRL algorithms (MIP-TD3, MIP-DDPG, MIP-SAC). Results shown in Fig. 3.4 are obtained as an average of over five algorithm executions. The average total reward increases rapidly during the training, while simultaneously, the number of voltage magnitude violations decreases. This is a typical training trajectory of penalty-based DRL algorithms. At the beginning of the training process, the DNN's parameters are randomly initialized, and as a consequence, the actions defined cause a high number of voltage magnitude violations. Throughout the training, introducing a large magnitude penalty term in the reward definition in (3.14) leads to updating the DNN's parameters, resulting in higher

quality actions, primarily learning to reduce voltage magnitude violations, and, later on, improving the general performance. All three MIP-DRL algorithms converged at around 1000 episodes. The total reward of MIP-TD3 and MIP-DDPG converged at 2.01 ± 0.02, and 1.94 ± 0.02, respectively. Compared to MIP-TD3 and MIP-DDPG, that of MIP-SAC converged to a low value, at 1.57 ± 0.01, indicating that MIP-SAC has a lower quality of actions. Notice that for MIP-DDPG and MIP-TD3, the operation cost significantly increases during the training process, while SAC does not improve after 400 episodes (Fig. 3.4). After the last training episode, the number of voltage magnitude violations of the MIP-TD3 algorithm is around 1. In contrast, a higher number of violations for the MIP-DDPG and MIP-SAC algorithms was observed at around 2. This result shows that DRL algorithms can effectively learn from interactions, reducing the number of voltage magnitude violations while minimizing the total operation cost by learning to dispatch the ESSs correctly. However, these trained policies *cannot* strictly enforce voltage magnitude constraints. If such algorithms are used directly in online execution phase, they might lead to infeasible operation, causing voltage violations.

3.5.3. Constraint Enforcement Capabilities and Performance

Figure 3.5 displays the voltage magnitude of the nodes in which the ESSs are connected and the ESSs' SOC during a typical day in the test dataset. The results showed in Fig. 3.5 are obtained after using the dispatch decisions provided by the MIP-DDPG, MIP-TD3 and MIP-SAC algorithms. Fig. 3.5(a) shows the voltage magnitude of the nodes in which the ESSs are connected, but in this case, disregarding their operation (i.e., ESSs are neither charging nor discharging), while Fig. 3.5(b) shows the day-ahead electricity price of that test day. As can be seen in Fig. 3.5(a), if the ESSs' operation is disregarded, the voltage magnitude at node 27 faces undervoltage problems between 14:00-16:00 and 18:00-20:30. Thus, a proper dispatch of the available ESSs must enforce that such voltage magnitude constraints are met. This is the case when executing the dispatch decisions provided by the developed algorithms, as shown in Fig. 3.5(c) for the MIP-DDPG algorithm, in Fig. 3.5(e) for the MIP-TD3 algorithm and in Fig. 3.5(g) for the MIP-SAC algorithm. As all the developed MIP-DRL algorithms dispatch the ESS connected at node 27 in discharging mode during the above-mentioned periods, all undervoltage issues are solved. In terms of dispatch decisions, and as seen in Fig. 3.5(d), (f) and (h), all the developed MIP-DRL algorithms learn to first discharge all ESSs to the minimum SOC during the period between 00:00 and 06:00. Then, all ESSs are dispatched in charging mode during the period between 10:00 and 17:00, when the price is low, to then operate in discharging mode during the period between 16:00-22:00. This operational schedule during the peak consumption period reduces the amount of power the external grid provides while simultaneously solving the undervoltage issues. Compared to the MIP-DDPG algorithm, the MIP-TD3 and MIP-SAC algorithms provide more conservative dispatch decisions, leading to higher operational costs. The operational cost resulting from the dispatch defined by the MIP-DDPG algorithm is 13.87 k€, 3.1% lower and 7.5% lower, than the dispatch defined by MIP-TD3 and MIP-SAC algorithms, respectively.

3.5.4. Performance Comparison with Benchmarks

Figure 3.6 further displays the charge/discharge decisions and SOC changes of the ESS connected to node 27 and defined by the MIP-DDPG, Safe DDPG, and standard DDPG

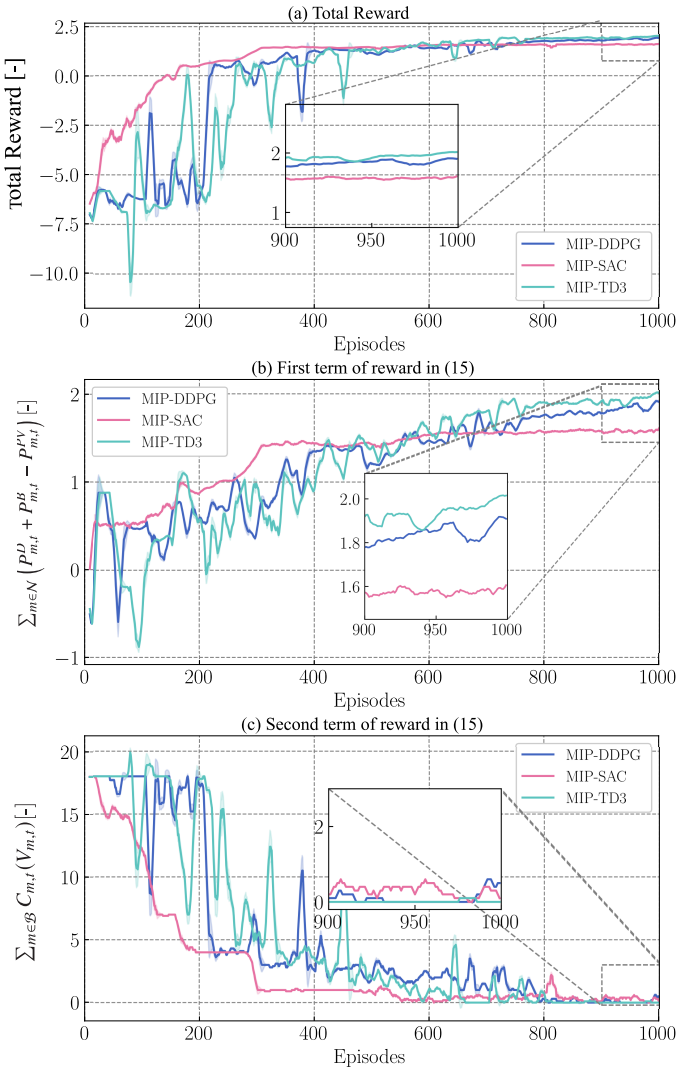


Figure 3.4: (*a*) Average total reward as in (3.15). (*b*) Operational cost or first term of reward in (3.15). (*c*) Cumulative penalty for voltage magnitude violations or second term of reward in (3.15), all during training.

algorithms, as well as the optimal solution provided by solving the NLP formulation considering the perfect forecast. Compared with the decisions defined by the NLP formulation, the dispatched decisions provided by the MIP-DDPG algorithm (Fig. 3.6(*c*))) shown a similar operation pattern, especially in the afternoon when the price changes dynamically (see Fig. 3.5(*b*)). As expected, when the electricity price is low, between 10:00-14:30, the MIP-DDPG algorithm dispatches the ESS in charging mode, while between 17:00 to 22:00, when the electricity price is high, the ESS is dispatched in discharging mode. In this sense, both algorithms, standard DDPG and Safe DDPG, captured and

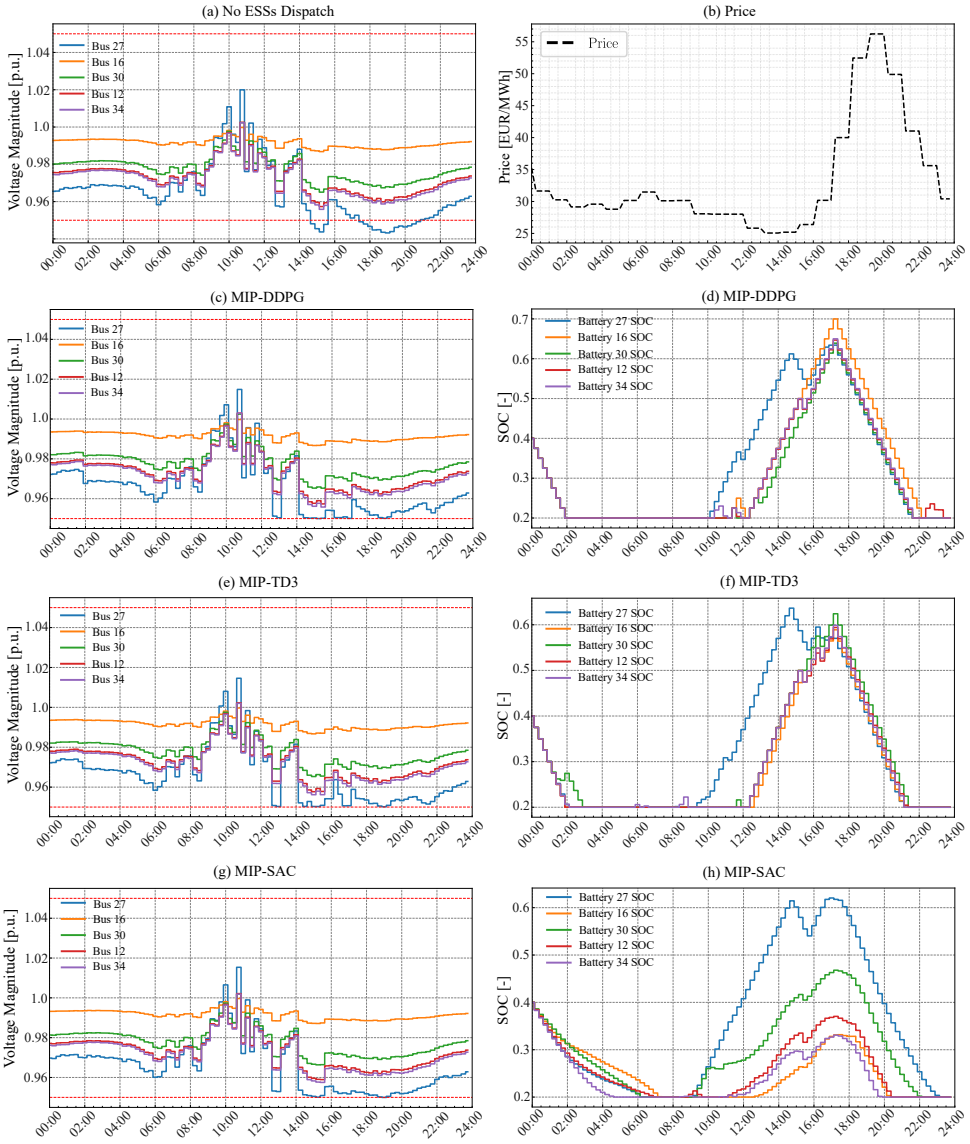


Figure 3.5: (a): Voltage magnitude for nodes in which the ESSs are connected, disregarding their operation. (b): Price in €/MWh. Voltage magnitude ((c), (e) (g)) in which the ESSs are connected and SOC of ESSs ((d), (f), (h)), after executing the dispatch decisions provided by the MIP-DDPG, MIP-TD3, and MIP-SAC algorithms, respectively.

exploited such arbitrage opportunities. Although the proposed MIP-DDPG algorithm failed to capture such behavior for this specific ESS, the decisions defined for the remaining ESSs ensured a maximization of profits without voltage magnitude violations. In this case, the cost of the dispatch decisions defined by the standard DDPG and safe DDPG

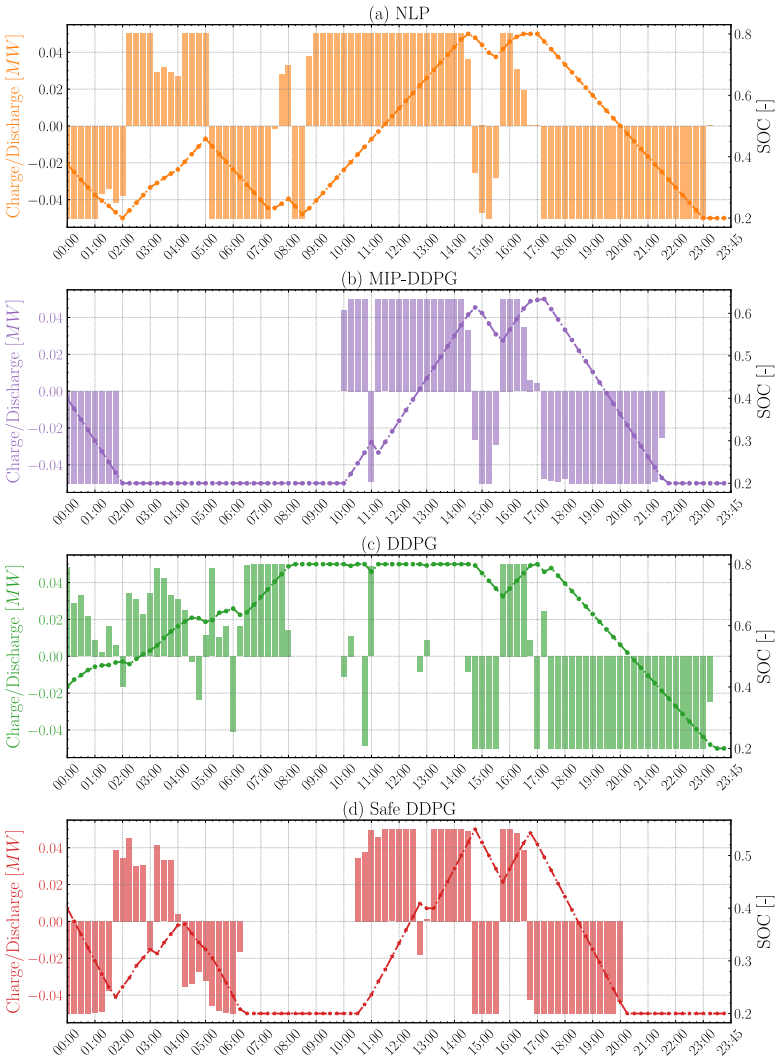


Figure 3.6: Dispatch decisions obtained by solving the NLP formulation (with perfect forecast) (a), and the ones provided by the MIP-DDPG (b), DDPG (c) and Safe DDPG (d) algorithms for the ESS connected to node 27.

algorithms are 22.3% and 27.3% higher, respectively, than the ones defined by their MIP counterpart. This shows that the standard DDPG and the Safe DDPG algorithm failed to fully leverage (and coordinate) all ESSs connected to the distribution network.

Figure 3.7 displays the voltage magnitude of node 27 in which an ESS is connected. The voltage magnitude shown is obtained after executing the ESSs dispatch decisions provided by MIP-DDPG, DDPG, and Safe DDPG algorithms, as well as the optimal solution provided by solving the NLP formulation and the operation without ESS dispatch. Without ESSs dispatch, node 27 suffers serious undervoltage conditions between 14:30-

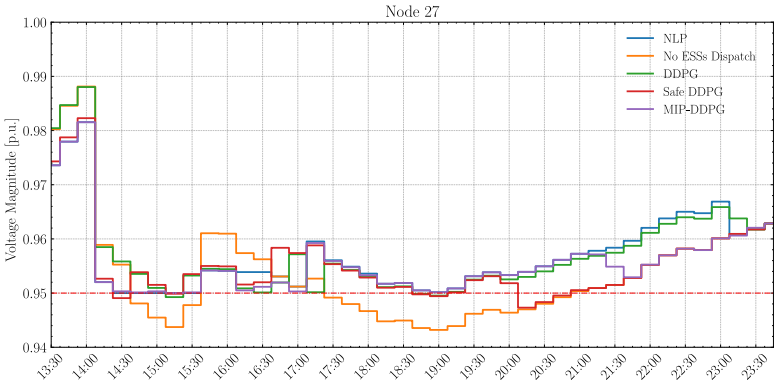


Figure 3.7: Voltage magnitude for node 27 after executing the dispatch decisions provided by the MIP-DDPG, Safe DDPG, and standard DDPG algorithms, as well as the optimal solution provided by solving the NLP formulation and the operation without ESS dispatch.

15:30 and 17:30-21:00 due to overloading. As expected, decisions defined by the MIP-DDPG algorithm strictly enforced the voltage magnitude constraints due to the feasibility guarantee feature of the proposed algorithm. In contrast, although the dispatch decisions defined by the standard DDPG and Safe DDPG algorithms can significantly alleviate the under-voltage condition, they fail to enforce voltage magnitude constraints in several time slots, such as between 18:30-19:30 and 20:00-21:00. These results indicate that the constraint enforcement capabilities of both these algorithms are not capable of handling complex stochastic environments (such a distribution network), and even the projection layer deployed by the Safe DDPG algorithm fails to map the relationship between actions and constraints accurately, ultimately deploying unfeasible actions.

Comparing the optimal solution obtained by solving the NLP formulation (shown in Fig. 3.6), with the solution provided by the proposed MIP-DRL algorithms, it can be seen that the MIP-DRL algorithms dispatched the ESSs following a more conservative approach (see charging/discharging behavior in Fig. 3.5(d)). The MIP-DRL algorithms generally avoid charging all ESSs to the maximum SOC when the electricity price is low. This can be considered a sub-optimal decision. In this case, the operational cost resulting from the dispatch defined by the MIP-DDPG, MIP-TD3 and MIP-SAC are 9.5%, 12.9%, and 18.4% higher, respectively, than the dispatch defined by the NLP formulation. The difference in this dispatch decision can be due to the estimated action-value function, which might not be good enough to represent the true action-value function. As the proposed MIP-DRL algorithms choose actions that maximize its Q-value estimation, the largest Q-value might not represent the best action for this specific state-action pair. Nevertheless, even in executing a sub-optimal decision, the proposed MIP-DRL algorithms enforce all voltage magnitude constraints, guaranteeing operational feasibility. On the other hand, the safe layer-based DRL algorithm i.e., Safe DDPG algorithm fails to enforce voltage magnitude constraints strictly, as the safety layer cannot track the dynamics of complex environments.

Table 3.3: Performance comparison of different DRL algorithms in an unseen test set of 30 days.

Algorithms	Operation Cost Error [%]	Voltage Magn. Violations [-]	Exec. Time [s]
MIP-TD3	13.2 ± 0.5%	0	57±6.7
MIP-DDPG	**10.4±0.7%**	0	43±5.1
MIP-SAC	19.3±1.5%	0	57±6.3
TD3	28.5±0.4%	33±2	16±0.1
DDPG	34.3±0.7%	45±11	16±0.1
SAC	32.2±0.5%	44±17	16±0.1
Safe-DDPG	39.7±0.8%	41±1	37±0.1

3.5.5. ERROR ASSESSMENT AND COMPUTATIONAL PERFORMANCE

Table 3.3 presents the average total error (with respect to the solution obtained by solving the NLP formulation with perfect forecast) for the operational cost, the average number of voltage magnitude violations, and the total average computational time of the proposed MIP-DRL algorithms as well as benchmark DRL algorithms, over 30 (unseen) test days. As can be seen in Table 3.3, the proposed MIP-TD3, MIP-DDPG, and MIP-SAC algorithms can strictly enforce the voltage constraints. Among all these MIP-DRL algorithms, MIP-DDPG has the lowest average error, 10.4%. In contrast, their standard counterparts, such as DDPG, showed poor performance reaching an error of 34.3%, and violating the voltage magnitude constraint in around 45 time steps. As expected, the computational time required to execute the proposed MIP-DRL algorithms is higher than standard DRL algorithms. This increase in the computational time results from the MIP formulation needed to be solved to enforce all the operational constraints (see (3.22)). Nevertheless, for this case, the proposed MIP-DRL algorithms can still be used for real-time operation as it only requires less than 60 seconds for one day (96 time-steps) execution.

3.5.6. SCALABILITY ANALYSIS

Table 3.4: Performance and Computational Effort of MIP-DDPG on Different Network Sizes

Nodes	Training Time [h]	Exec. Time [s]	Voltage Magn. Violations	Operation Cost Error (%)
34	4	43±5.1	0	10.4±0.7
69	4.7	49±6.9	0	10.1±0.9
123	6.5	53±3.4	0	11.3±0.7

Table 3.4 presents the performance and computational effort of the MIP-DDPG algorithm on different sizes of distribution networks. Table 3.4 includes the training time, execution time, number of voltage magnitude violations, and operation cost error for networks with 34, 69, and 123 nodes. The training time increases with the size of the distribution network, as expected. For instance, training on a 34-node network takes 4 hours, while training on a 123-node network takes 6.5 hours. This increase is primarily due to the time required to solve the power flow equations during the training process. As the network size grows, the complexity of solving these equations increases, leading to longer training times. The operation cost error, remains consistent across differ-

ent network sizes, with errors ranging from 10.1% to 11.3%. This suggests that the size of the distribution network does not significantly impact the performance of the MIP-DDPG algorithm. Moreover, MIP-DDPG successfully enforces voltage constraints across all tested network sizes, as evidenced by the absence of voltage magnitude violations. Finally, the execution time does not increase significantly with the network size as is shown in Table 3.4. The execution time for the 34-node network is 43 seconds, while for the 123-node network, it is 53 seconds. This is because the execution time is primarily influenced by the size of the Q-network used in the MIP formulation rather than the size of the distribution network. Once the Q-network is trained, the execution phase involves solving the MIP, which is not directly dependent on the distribution network size but on the complexity of the Q-network.

3.6. DISCUSSION

We have successfully combined deep learning and optimization theory to bring constraint enforcement to DRL algorithms. By using the trained Q-network as the surrogate function of the optimal operational decisions, we have guaranteed the optimality of the action from the Q-network through the formulated MIP. Moreover, by integrating the voltage constraints into the formulated MIP, the feasibility of the action is enforced. However, the performance of MIP-DRL algorithms is determined by the approximation quality of the Q-network, obtained after the training process is performed. During this training process, the Q-iteration faces the exploration vs. exploitation dilemma, which can impact the approximation quality. For instance, the MIP-DDPG algorithm outperforms the MIP-TD3 algorithm, while the MIP-SAC algorithm performs poorly in the framework. This discrepancy may be caused by the divergence between the exploration policies leading to different exploration efficiencies and Q-networks update rules. The conservative performance of the MIP-SAC algorithm might be caused by the soft Q updating rule, which introduces more assumptions, impacting the estimation for accurate approximation.

Formulating a trained Q-network as a MIP problem introduces extra computation time due to the maximization of the Q-value function. In this case, such a MIP formulation is considered to be an NP-Complete problem. The worst-case computation time grows exponentially with the number of integer variables, which is proportional to the total number of ReLU activation functions used. However, the computation time can be greatly reduced by various techniques like improved branch-and-bound, and customized ReLU function algorithms, developed in recent years [116]. Previous research results show that only 0.8 seconds are needed for solving a MIP formulated by a network with 300 ReLU units with an excellent CPU [117]. In our experiments, the proposed MIP-DRL algorithms required less than 60 seconds for execution, supporting the applicability of DRL algorithms in real systems. In summary, the proposed MIP-DRL algorithms can provide good quality dispatch decisions while *strictly* enforcing all voltage magnitude constraints, leading to high-quality feasible decisions. Compared to standard DRL algorithms, this superiority is achieved by directly transforming the Q-network (after training) as a MIP formulation, used to define the optimal solution instead of leveraging an approximated policy; while operational constraints are added on top of the obtained MIP formulation, guaranteeing feasibility.

4

DistFlow Safe Reinforcement Learning Algorithm for Voltage Magnitude Regulation in Distribution Networks

The integration of distributed energy resources (DER) has escalated the challenge of voltage magnitude regulation in distribution networks. Model-based approaches, which rely on complex sequential mathematical formulations, can not meet real-time demand. Deep reinforcement learning (DRL) offers an alternative by utilizing offline training with distribution network simulators and then execution without online computation. However, DRL algorithms fail to enforce voltage magnitude constraints during training and testing, potentially leading to serious operational violations. To tackle these challenges, we introduce a novel safe reinforcement learning algorithm, the DistFlow Safe Reinforcement Learning (DF-SRL), designed specifically for real-time voltage magnitude regulation in distribution networks. The DF-SRL algorithm incorporates a DistFlow linearization to construct an expert knowledge-based safety layer. Subsequently, DF-SRL overlays this safety layer on top of the agent's policy, recalibrating unsafe actions to safe domains through a quadratic programming formulation. Simulation results show the proposed DF-SRL consistently ensures voltage magnitude constraints during the training and real-time operation (test) phases, achieving faster convergence and higher performance, setting it apart from (safe) DRL benchmarks.

Parts of this chapter have been published in IEEE Journal of Modern Power Systems and Clean Energy with the title: *DistFlow Safe Reinforcement Learning Algorithm for Voltage Magnitude Regulation in Distribution Networks*, doi: 10.35833/MPCE.2024.000253. [118].

4.1. Introduction

Distribution networks have experienced a notable increase in distributed energy resources (DER) integration, including residential PV systems, energy storage systems (ESSs), and plug-in electric vehicles (EV) [119]. This rise in DERs contributes to sustainability efforts and poses operational challenges to distribution system operators (DSOs). Among these challenges, voltage magnitude regulation has surfaced as a predominant concern [120]. Aggregators, who control various DERs, have stepped in to offer a solution. By providing significant flexibility to DSOs, aggregators enable the strategic procurement and deployment of this flexibility, thereby facilitating efficient voltage regulation [121].

Implementing voltage magnitude regulation adopts one of two approaches: model-based and model-free approaches. Model-based approaches manage voltage magnitude regulation by solving mathematical formulations defined via an objective function and a set of operational constraints [122]. However, the intricacy of these model-based approaches increases with the complexity of distribution networks and sequential regulation slots because they necessitate complete network and DER information. Therefore, solving such formulations can be computationally intensive and thus can not meet real-time demand [123]. Conversely, model-free deep reinforcement learning (DRL), represents an alternative approach that does not require online computation by leveraging an offline training procedure and distribution network simulators [124]. Nevertheless, a significant drawback of such DRL algorithms is their inability to ensure action feasibility and, thus, safety [125, 126]. To address this, some studies have formulated the voltage magnitude constraint as a soft constraint, i.e., a fixed [127] or trainable penalty term [107], added to the reward function and used to guide the DRL algorithm during training. For instance, the RL algorithm proposed in [128] follows this approach, developed to define the ESSs schedule to minimize operational costs while respecting voltage magnitude limits. Nevertheless, this approach fails to enforce such constraints strictly during training and real-time operation.

Several safe DRL approaches have recently been developed to enforce operational constraints in control systems [129]. In [108], a constraint Soft Actor-Critic (SAC) algorithm was developed for EV charging in residential microgrids to cater to the increasing prominence of EVs. Using a constrained MDP formulation and a ladder electricity pricing scheme, this approach showed promising results in reducing action space dimensionality and ensuring safe EV charging. Another study [107] implemented primal-dual optimization within a safe RL framework, showing superior performance in terms of energy cost minimization and constraint adherence. In [98], a safe DRL algorithm was introduced to define a fast-charging strategy for lithium-ion BES to enhance the efficiency of EV charging without compromising BES safety. Utilizing the SAC-Lagrange DRL within a cyber-physical system framework; the strategy optimizes charging speeds by leveraging an electro-thermal model, outperforming existing DDPG-based and SAC-based DRL methods in terms of optimality.

To ensure that the updated policy stays within a feasible set, in [99, 100], a cumulative constraint violation index is kept below a predetermined threshold. This approach was also used in [74, 75], in which the constraint violation index is designed to reflect the voltage and current magnitude violation level due to the ESSs dispatch defined. Nevertheless, this constrained policy was initially developed to handle cumulative or chance

constraints after training [100]. On the contrary, voltage magnitude violation issues in distribution networks are state-wise constraints, which do not rely on historical trajectories or random variables but hinge on the current state of the environment [130]. Consequently, applying constrained policy optimization methods to voltage regulation issues can not offer a probabilistic sense of safety. In [105], the trained DRL algorithm is formulated as a mixed-integer programming (MIP) formulation and voltage magnitude constraints are added to the MIP. By solving this extended MIP, the actions from the DRL algorithm are projected to safe action spaces strictly enforcing constraints. Nevertheless, this approach can not meet the real-time operation requirements if the formulated MIP becomes too large. In [131], the stability of distribution network controlled by DRL algorithms is guaranteed if the system adheres to specific Lipschitz constraints. However, formulating such Lipschitz sets for distribution networks is quite changeable. In [132], a constrained-SAC algorithm is proposed to address Volt-Var control challenges. constrained-SAC combines the maximum-entropy framework, the method of multiplier, a device-decoupled neural network structure, and an ordinal encoding scheme to achieve scalability, sample efficiency, and constraint satisfaction. However, the algorithm can only be applied to discrete-action problems.

Safe layer-based DRL algorithms are suitable to handle the state-wise constraints (i.e., voltage magnitude), which formulate a policy-independent safe layer to project actions defined by DRL algorithms into a feasible set. In [133], a DNN-assisted projection-based DRL method is proposed for the safe control of distribution networks. This approach leverages a pre-trained DNN to accelerate the projection calculations, enabling the rapid identification of safe actions. However, a critical limitation of this method is the reliability of the safe actions produced by the DNN, since the DNN is trained on historical data, the quality and representativeness of this data are paramount. Alternatively, a linear safe layer is trained by the data collected from a random policy with the environment [85]. In [70], a safety layer is built upon DRL algorithms to filter out unsafe actions before the execution, while voltage magnitude is enforced by solving projection. A similar approach was implemented in [103] to regulate the distribution networks' voltage magnitude via controlling smart transformers. Yet, these approaches mainly rely on training a linear safety layer to first capture the sensitivity between station-action pair and constraint violations, and then filter out unsafe actions before they are executed. Therefore, the safety guarantee performance for these approaches is highly dependent on the quality of the trained linear safe layer. Given the complex relationships between system dynamics and multi-dimension constraints involved in voltage regulation problems, training such a linear safety layer often proves to be a significant challenge [85]. Consequently, the trained safety layer can rarely provide a safety guarantee for the voltage magnitude regulation problem in the distribution network, leading to sub-optimal performance and violations.

Drawing on the pivotal insights [134, 135, 136] that integrating expert knowledge can significantly enhance safety and agent performance, we introduce the DistFlow Safe Reinforcement Learning (DF-SRL) algorithm. This is the first effort to tackle state-wise voltage regulation issues in distribution networks by applying DRL strategies, augmented with an expert-knowledge-based safety layer. This innovation addresses existing gaps in voltage regulation research through several key contributions:

- The proposed DF-SRL algorithm incorporates a DistFlow linearization to devise a safety layer, leveraging expert knowledge insights to accurately map the relationship between the agent's actions and voltage magnitude variations in distribution networks.
- The DRL algorithm overlays the safe layer on top of the DRL policy to recalibrate potentially unsafe actions to conform to safe parameters by optimizing the proximity of these actions in Euclidean space.
- The error of the safe layer introduced by linearization is corrected by the slack parameter, and a detailed sensitivity and scalability analysis is conducted.
- The proposed DF-SRL algorithm ensures the practicality and real-time viability of actions and guarantees safety constraints during both the training and application phases.

4.2. Voltage Magnitude Regulation Problem

Voltage fluctuations in distribution networks are predominantly due to variations in active power, such as from overload conditions or high inflows from photovoltaic (PV) systems [137]. These fluctuations are more directly linked to active power changes, affecting voltage magnitude significantly. By focusing on active power, aggregators can utilize DERs like battery storage and controllable loads more effectively. This aligns with operational strategies that maximize the impact of available resources while ensuring compliance with safety and reliability standards.

The voltage magnitude regulation framework is depicted in Fig. 4.1. Each network node is associated with an aggregator that oversees a group of consumers with DERs. These aggregators are empowered to fully control the DERs of their designated consumers, playing a pivotal role in the dynamic management of the distribution network. Aggregators collect consumer data, build baseline electrical consumption profiles, and share the active power flexibility with the DSO control center. Subsequently, the DSO control center deploys a voltage magnitude regulation algorithm to determine the required active power flexibility each aggregator must provide.

In this paper, we focus on developing an RL-based algorithm to assist the DSO control center in accurately determining the required flexibility provision of each aggregator to achieve voltage magnitude regulation.

4.2.1. Mathematical Programming Formulation

In general, the voltage magnitude regulation problem can be modeled using the nonlinear programming (NLP) formulation given by (4.1)–(4.9). The objective function in (4.1) aims to minimize the use of flexible active power $p^B_{m,t}$ provided by all aggregators within the set $m \in \mathcal{N}$, aiming to regulate the voltage magnitude over the time horizon $\mathcal{T}$.

$$\min_{p^B_{m,t}} \left\{ \sum_{t\in\mathcal{T}} \left[\sum_{m\in\mathcal{N}} \left(\| p^B_{m,t} \| \right) \Delta t \right] \right\}, \tag{4.1}$$

Subject to:

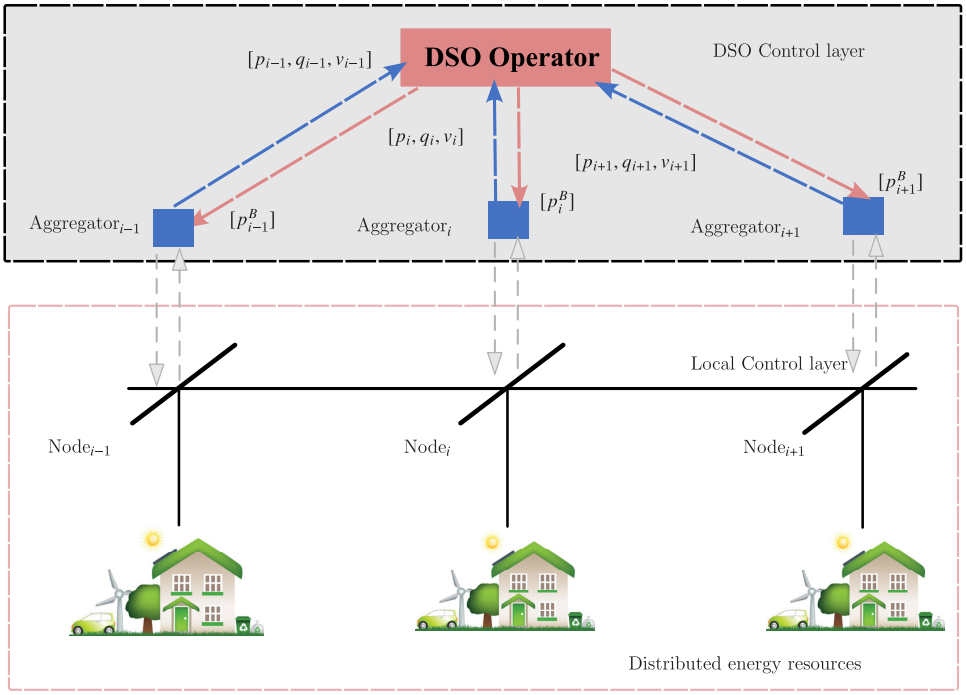


Figure 4.1: Voltage magnitude regulation framework for DSO and aggregators. These aggregators oversee groups of consumers equipped with distributed energy resources (DERs) such as residential PV systems, battery energy storage systems (BESS), and plug-in electric vehicles (EVs).

$$\sum_{nm\in\mathscr{L}} p_{nm,t} - \sum_{mn\in\mathscr{L}} (p_{mn,t} + r_{mn} i^2_{mn,t}) + p^B_{m,t} + p^{PV}_{m,t} + p^S_{m,t} = p^D_{m,t} + p^{EV}_{m,t} \quad \forall m \in \mathscr{N}, \forall t \in \mathscr{T} \tag{4.2}$$

$$\sum_{nm\in\mathscr{L}} q_{nm,t} - \sum_{mn\in\mathscr{L}} (q_{mn,t} + x_{mn} i^2_{mn,t}) + q^S_{m,t} = q^D_{m,t} \quad \forall m \in \mathscr{N}, \forall t \in \mathscr{T} \tag{4.3}$$

$$v^2_{m,t} - v^2_{n,t} = 2(r_{mn} p_{mn,t} + x_{mn} q_{mn,t}) + (r^2_{mn} + x^2_{mn}) i^2_{mn,t} \quad \forall m,n \in \mathscr{N}, \forall t \in \mathscr{T} \tag{4.4}$$

$$v^2_{m,t} i^2_{mn,t} = p^2_{mn,t} + q^2_{mn,t} \quad \forall m,n \in \mathscr{N}, \forall t \in \mathscr{T} \tag{4.5}$$

$$\underline{p}^B_{m,t} \le p^B_{m,t} \le \overline{p}^B_{m,t} \quad \forall m \in \mathscr{N}, \forall t \in \mathscr{T} \tag{4.6}$$

$$\underline{v}^2 \le v^2_{m,t} \le \overline{v}^2 \quad \forall m \in \mathscr{N}, \forall t \in \mathscr{T} \tag{4.7}$$

$$0 \le i^2_{mn,t} \le \overline{i}^2_{mn} \quad \forall mn \in \mathscr{L}, \forall t \in \mathscr{T} \tag{4.8}$$

$$p^S_{m,t} = q^S_{m,t} = 0 \quad \forall m \in \mathscr{N}\backslash\{1\}, \forall t \in \mathscr{T} \tag{4.9}$$

The distribution network is formulated based on the power flow formulation shown in (4.2)–(4.5), according to the active power $p_{mn,t}$, reactive power $q_{mn,t}$ and current magnitude $i_{mn,t}$ of lines, and the voltage magnitude $v_{m,t}$ of nodes. The expression in (4.6)

enforces the used flexible active power within the boundaries that each aggregator provides, while (4.7) and (4.8) enforce the voltage magnitude and line current limits, respectively. Finally, (4.9) enforces that only one node is connected to the substation.

4.2.2. CMDP Formulation

The voltage magnitude regulation problem can be modeled as a case of constrained Markov decision processes (CMDP), represented by a 6-tuple $(\mathscr{S}, \mathscr{A}, \mathscr{P}, \mathscr{R}, \gamma, \mathscr{C})$. Here, $\mathscr{S}$ represents a state space encompassing the power system's observable states, $\mathscr{A}$ denotes an action space representing the possible control actions, $\mathscr{P}$ is the state transition probability function capturing the system's dynamics, $\mathscr{R}$ is the reward function guiding the optimization, γ is a discount factor reflecting the importance of future rewards, and $\mathscr{C}$ constitutes a set of immediate-constraint functions ensuring operational safety and feasibility. The decision as to which action a_t is chosen in a certain state s_t is governed by a policy $\pi(a_t|s_t)$. The agent employs the policy to interact with the formulated CMDP and define a trajectory of states, actions, and rewards: $\tau = (\boldsymbol{s}_0, \boldsymbol{a}_0, \boldsymbol{s}_1, \boldsymbol{a}_1, \cdots)$. This trajectory not only aims to maximize the cumulative reward but also adheres to the system constraints, thereby balancing the objectives of operational efficiency and safety.

State

The state at time t encapsulates the distribution network's current operational status, providing a comprehensive view of the system's dynamics, and it is defined by:

$$s_t = (p^N_{m,t}, v_{m,t}, \underline{p}^B_{m,t}, \overline{p}^B_{m,t}|_{m\in\mathcal{N}}), \tag{4.10}$$

where $p^N_{m,t} = p^D_{m,t} - p^{PV}_{m,t} - p^{EV}_{m,t}$ corresponds to the nodal net active power, capturing the balance between demand, photovoltaic generation, and electric vehicle consumption at each node m. $v_{m,t}$ represents the voltage at each node m before control; $\underline{p}^B_{m,t}$ and $\overline{p}^B_{m,t}$ are flexibility boundary can be provided by aggregator connected to m_{th} node[1].

Action

The action space $\mathscr{A}$ consists of the set of all possible active power adjustments at each node m, defined as $\mathscr{A} = \{a_t \mid a_t = p^B_{m,t}, \underline{p}^B_{m,t} \le p^B_{m,t} \le \overline{p}^B_{m,t}, \forall m \in \mathcal{N}\}$.

Reward

The DSO seeks to regulate voltage magnitude into defined boundaries while minimizing the use of total active power flexibility provided by aggregators. Thus, the reward function r_t is defined as the negative of the total used flexible active power, as next:

$$r_t = -\sum_{m\in\mathcal{N}} \|p^B_{m,t}\| \tag{4.11}$$

This formulation incentivizes the minimization of the total active power flexibility utilized, thereby promoting energy efficiency and cost-effectiveness in voltage regulation. Given the state s_t and action a_t at time step t, the system transit to the next state s_{t+1} defined by the transition probability function that can be expressed as:

[1] Flexibility for voltage regulation at each aggregator can vary over day and time slots [138].

$$p(S_{t+1}, R_t | S_t, A_t) = \Pr\{S_{t+1} = s_{t+1}, R_t = r_t \mid S_t = s_t, A_t = a_t\}. \tag{4.12}$$

The RL agent's goal is to find a policy that maximizes the cumulative discounted return $J(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{T} \gamma^t r_t\right]$ while ensuring no constraint is violated during the exploration and exploitation processes. The penalty term induced by the constraint violations $C_{m,t}(\pi)$ denotes the voltage magnitude violation of the m_{th} node at step t, which is defined as:

$$C_{m,t} = \max\left\{0, \left|v_0 - v_{m,t}\right| - \frac{\overline{v} - \underline{v}}{2}\right\}, \forall m \in \mathcal{N}. \tag{4.13}$$

Here, v_0 represents the voltage magnitude at the reference or slack node, which is typically considered constant and known. This formulation ensures that $C_{m,t}$ represents a positive penalty term when the voltage magnitude at node m deviates outside the acceptable range defined by $\underline{v}$ and $\overline{v}$, and is zero otherwise.

The voltage magnitude regulation problem formulated as a CMDP can then be expressed using the next constrained optimization formulation:

$$\begin{aligned} &\max_{\pi} = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\mathcal{T}} \gamma^t r_t\right] \\ &\text{s.t. } C_{m,t}(\pi) = 0, \forall m \in \mathcal{N}, \forall t \in \mathcal{T} \end{aligned} \tag{4.14}$$

In this formulation, $C_{m,t}$ serves as a constraint in the CMDP, ensuring that the policy π leads to actions that maintain the voltage magnitude within the specified limits. It is indirectly influenced by the policy through its impact on the state s_t and the action a_t.

4.3. PROPOSED DISTFLOW SAFE RL ALGORITHM

The proposed algorithm is defined through a parameterized policy network, denoted by $\pi_\omega(\cdot)$. This policy network selects actions based on the current state, performing exploration and exploitation. To enhance safety and ensure that voltage magnitude constraints are met during the exploration, we introduce a safety layer on top of the policy network $\pi_\omega(\cdot)$. A safety layer is designed based on the parameters and topology of the distribution network, enabling a projection of the original action proposed by the RL algorithm onto a safe domain. A more detailed explanation is provided next.

4.3.1. DEEP DETERMINISTIC RL ALGORITHMS

Traditional value-based DRL algorithms fail to solve the voltage magnitude regulation problem due to the continuous nature of the state and action spaces [44]. Alternatively, Deep Deterministic Policy-based DRL algorithms, such as DDPG [23], and TD3 [24], are capable of handling continuous actions by simultaneously maintaining a policy (actor) $\pi_\omega(s_t)$, used to sample actions, and a trained Q-function (critic) $Q_\theta(s_t, a_t)$, used to guide the update direction of the policy network. The TD3 algorithm is an improved version of the DDPG algorithm, which uses two Q-networks and delayed critic network improvement to reduce the overestimation bias of the critic network in DDPG. In general, the

TD3 algorithm updates the actor-network as

$$\omega \leftarrow \omega + \nabla_\omega \frac{1}{|B|} \sum_{s_t \in B} \left(\min_{i=1,2} \{ Q_{\theta_i}(s_t, \pi_\omega(s_t)) \} \right), \tag{4.15}$$

while the critic update iteration is defined as

$$\min_\theta \sum_{s \in B} \left(r_t + \gamma \min_{i=1,2} \{ Q_{\theta_i^{\text{target}}}(s_{t+1}, \pi_\omega(s_{t+1})) \} - Q_{\theta_i}(s_t, a_t) \right)^2 \tag{4.16}$$

Although the TD3 algorithm effectively handles continuous action space problems, it can not enforce constraints during the training and testing. To solve the CMDP formulation using the TD3 algorithm, the constraint violation functions $C_{m,t}$ should be added as penalty term to the reward function in (4.11), defined as

$$r_t = - \sum_{m \in \mathcal{N}} \| p^B_{m,t} \| - \sigma \left[\sum_{m \in \mathcal{N}} C_{m,t} \right], \tag{4.17}$$

where σ is used to balance the weights between the total required flexibility and the penalty incurred by the voltage magnitude violations. The constrained optimization problem is reformulated into an unconstrained one in this procedure. However, directly applying penalty terms to the reward function cannot guarantee the feasibility strictly, leading to infeasible operations and poor performance [139]. To overcome this, we introduce a linear safety layer on the top of the TD3 algorithm to ensure the feasibility of committed actions during the training and testing procedures, as explained in the next section.

4.3.2. Linear Power Flow Formulation

Given the topology of a distribution network, the incidence matrix $\boldsymbol{M_0}$ can be defined by:

$$\boldsymbol{M_0} = \boldsymbol{F} - \boldsymbol{T} = [\boldsymbol{m_0}, \boldsymbol{M}] \tag{4.18}$$

where,

$$\begin{aligned} [\boldsymbol{F}]_{\ell,i} &= \begin{cases} 1 & f(\ell) = i, \ell \in \mathcal{L}, i \in \mathcal{N} \\ 0 & \text{otherwise} \end{cases} \\ [\boldsymbol{T}]_{\ell,j} &= \begin{cases} 1 & t(\ell) = j, \ell \in \mathcal{L}, j \in \mathcal{N} \\ 0 & \text{otherwise.} \end{cases} \end{aligned} \tag{4.19}$$

where $\boldsymbol{m_0}$ is the column corresponding to the slack node, and $[\boldsymbol{F}]_{\ell,i}$ and $[\boldsymbol{T}]_{\ell,j}$ are the connection matrix.

Given the diagonal matrix $D(\boldsymbol{r_{mn}})$ and $D(\boldsymbol{x_{mn}})$, as functions of the resistant vector $\boldsymbol{r_{mn}}$ and the reactance vector $\boldsymbol{x_{mn}}$, the relationship between the voltage magnitude of nodes, defined by vector $\boldsymbol{v_m}$, and the net active and reactive power injection, defined by vectors $\boldsymbol{p^N_m}$, $\boldsymbol{q^N_m}$, respectively, can be expressed as

$$\boldsymbol{M v^2_m} = \boldsymbol{M} v_0^2 \mathbf{1}_{|\mathcal{L}|} + 2[\mathrm{D}(\boldsymbol{r_{mn}}) \boldsymbol{B^T} \boldsymbol{p^N_m} + \mathrm{D}(\boldsymbol{x_{mn}}) \boldsymbol{B^T} \boldsymbol{q^N_m}] + \boldsymbol{C} \boldsymbol{c}^2 \tag{4.20}$$

where $\mathbf{1}_{|\mathscr{L}|}$ is the unit vector, and matrix $\boldsymbol{B}$ and $\boldsymbol{C}$, are defined as

$$\boldsymbol{B} = \left(\mathbb{I} - \boldsymbol{T}\boldsymbol{F}^{T}\right)^{-1} \tag{4.21}$$

$$\boldsymbol{C} = 2(\mathrm{D}(\boldsymbol{r_{mn}})\boldsymbol{B}\boldsymbol{D}(\boldsymbol{r_{mn}}) + \mathrm{D}(\boldsymbol{x_{mn}})\boldsymbol{B}\boldsymbol{D}(\boldsymbol{x_{mn}})) - \mathrm{D}\left(\boldsymbol{r_{mn}}^{2} + \boldsymbol{x_{mn}}^{2}\right) \tag{4.22}$$

The linear power flow formulation presented in (4.20) involves an approximation that neglects the quadratic term c^2, which represents the line losses in the distribution network. This simplification is based on the findings in [140], where it is argued that in most practical scenarios, especially in distribution networks, the line losses can be considered relatively small compared to the other terms in the power flow equations. Thus, the quadratic term c^2 in (4.20) is neglected, turning the expression linear in $\boldsymbol{v_m^2}$. This linear expression can further be used to derive a direct relationship between the action $\boldsymbol{a}$ vector, corresponding to the dispatch decision of the aggregators, i.e., $\boldsymbol{a} = [p^B_{1,t}, ...p^B_{m,t}, ...p^B_{|\mathscr{N}|,t}]$, and $\boldsymbol{v_m^2}$, as next

$$\boldsymbol{M}\boldsymbol{v^2} = \boldsymbol{M}v_0^2\mathbf{1}_{|\mathscr{L}|} + 2[\mathrm{D}(\boldsymbol{r_{mn}})\boldsymbol{B}\boldsymbol{T}(\boldsymbol{p_m^N} - \boldsymbol{a}) + \mathrm{D}(\boldsymbol{x_{mn}})\boldsymbol{B}\boldsymbol{T}\boldsymbol{q_m^N}]. \tag{4.23}$$

4.3.3. SAFETY LAYER FORMULATION

The relationship expressed in (4.20) is utilized to establish a linear mathematical programming formulation to project potentially unsafe actions, defined by the RL algorithm, into a secure operational region. The primary objective of this formulation is to find the nearest safe action $\hat{\boldsymbol{a}}$ that minimizes the Euclidean distance from the original potentially unsafe action $\boldsymbol{a}$. Thereby, ensuring minimal deviation from the intended control strategy while strictly adhering to operational and safety constraints. The safe action projection is achieved by solving the optimization problem:

$$\hat{\boldsymbol{a}} = \arg\min_{\hat{\boldsymbol{a}}} \frac{1}{2} \|\hat{\boldsymbol{a}} - \boldsymbol{a}\|^2. \tag{4.24}$$

Subject to:

$$(\boldsymbol{v_m^2}\mathbf{1}_{|\mathscr{L}|} + 2\boldsymbol{M}^{-1}[\mathrm{D}(\boldsymbol{r_{mn}})\boldsymbol{B}\boldsymbol{T}(\boldsymbol{p_m^N} - \hat{\boldsymbol{a}}) + \mathrm{D}(\boldsymbol{x_{mn}})\boldsymbol{B}\boldsymbol{T}\boldsymbol{q_m^N}]) \leq \overline{v}^2 - \epsilon \tag{4.25}$$

$$(\boldsymbol{v_m^2}\mathbf{1}_{|\mathscr{L}|} + 2\boldsymbol{M}^{-1}[\mathrm{D}(\boldsymbol{r_{mn}})\boldsymbol{B}\boldsymbol{T}(\boldsymbol{p_m^N} - \hat{\boldsymbol{a}}) + \mathrm{D}(\boldsymbol{x_{mn}})\boldsymbol{B}\boldsymbol{T}\boldsymbol{q_m^N}]) \geq \underline{v}^2 + \epsilon \tag{4.26}$$

The slack parameter ϵ is introduced to manage the relaxation conditions for the voltage magnitude limits, which compensates for the inaccuracies introduced by the linear model approximation of real voltage magnitudes. By incorporating ϵ, we allow for a buffer in the operational constraints that accommodates potential deviations between the predicted and actual voltage magnitudes. This ensures that the projected actions remain within safe operational boundaries, even when the linear relationship underestimates or overestimates the effects of control actions on the voltage levels.

4.3.4. PROPOSED DF-SRL ALGORITHM

The proposed safety layer can project action a_t to safe domains $\hat{a}_t$ during the training and online execution process. The proposed DF-SRL algorithm will update the actor and critic networks based on the collected safe trajectories $(s_t, \hat{a}_t, r_t, s_{t+1})$ in the replay buffer R. Therefore, DF-SRL redefined the actor-network iteration rule by:

$$\omega \leftarrow \omega + \nabla_\omega \frac{1}{|B|} \sum_{s_t \in B} \left(\min_{i=1,2} \{Q_{\theta_i}(s_t, \hat{a}_t)\} \right), \tag{4.27}$$

while the critic update iteration in (4.16) is redefined as

$$\min_\theta \sum_{s \in B} \left(r_t + \gamma \min_{i=1,2} \{Q_{\theta_i^{\text{target}}}(s_{t+1}, \hat{a}_t)\} - Q_{\theta_i}(s_t, \hat{a}_t) \right)^2. \tag{4.28}$$

Note that the developed DF-SRL algorithm to integrating the safe layer is specifically designed to be compatible with off-policy model-free algorithms. The off-policy nature of the DF-SRL algorithm allows it to learn from experiences generated by a behavior policy that differs from the target policy trying to learn. This characteristic is crucial for the integration of the safety layer, as it allows the algorithm to handle the mismatched distribution between the original actions, a_t, and the safe actions, $\hat{a}_t$, without impairing the update performance. Consequently, the safety layer can project potentially unsafe actions into a safe domain, ensuring operational feasibility while maintaining the integrity of the learning process. The DF-SRL algorithm maintains its model-free nature by not explicitly learning the state transition function of the constructed MDP [141].

In addition to the integration of the safety layer, DF-SRL introduces significant novelty in the policy iteration and interaction process. More than just filtering actions, the safety layer actively changes the nature of the interaction data that is fed back into the learning process of the RL agent. By modifying the actions before they are executed (and thus the resulting state transitions and rewards), the safety layer ensures that the data used for training is not only rich in terms of learning opportunities but also aligned with operational safety requirements. This leads to an improvement in both the performance and safety of the learned policy.

Algorithm 1 presents the step-by-step procedure of the proposed DF-SRL algorithm, while Fig. 4.2 illustrates the interaction of the actor and critic models with the environment during the training process. The training process begins by randomly initializing the parameters of the DNN functions Q_θ and $Q_{\theta^{\text{target}}}$, as well as defining the parameters of the safety layer i.e., $\mathrm{D}(\boldsymbol{r_{mn}}), \mathrm{D}(\boldsymbol{x_{mn}}), \boldsymbol{B}, \boldsymbol{T}, \boldsymbol{M}$. For each training epoch, at each time step t, the policy π_ω receives the state s_t and samples an action a_t. The safety layer then assesses whether the action a_t falls within the safe domain. The projection model is activated to project actions to a safe action, denoted as $\hat{a}_t$, only if action a_t could lead to voltage magnitude violations. Next, a transition tuple $(s_t, \hat{a}_t, r_t, s_{t+1})$ is compiled and stored in a replay buffer R. A subset B of these samples is subsequently selected and used to update the parameters of the functions Q_θ, $Q_{\theta^{\text{target}}}$, and π_ω as detailed in Algorithm 1. This iterative procedure continues until the maximum number of epochs is reached, ensuring that the RL agent can efficiently explore the action space without breaching voltage magnitude limits, thereby ensuring operational feasibility.

4.4. Simulation Results and Discussions

4.4.1. Simulations Setup, Data and Implementation

Algorithm 4: Proposed DF-SRL Algorithm

Define the maximum training epochs T, epoch length L.
Initialize parameters of functions Q_θ, $Q_{\theta^{\text{target}}}$, and π_ω; Initialize reply buffer R. ;
Define the parameters of the safety layer: $\mathrm{D}(\boldsymbol{r_{mn}}), \mathrm{D}(\boldsymbol{x_{mn}}), \boldsymbol{B}, \boldsymbol{T}, \boldsymbol{M}$.;
for $t = 1$ **to** T **do**
  Sample an initial state s_0 from the initial distribution **for** $l = 1$ **to** L **do**
    Sample an action with exploration noise $a_t \sim \pi_\omega(s_t) + \epsilon, \epsilon \sim \mathcal{N}(0, \sigma)$;
    if *not* $\underline{v} < v < \overline{v}$ **then**
      Projecting a_t to safe action $\hat{a}_t$ by solving (4.24) s.t. (4.25)–(4.26).
    else $\hat{a} = a$;
    Interact with the distribution network and observe the reward r_t and the new state s_{t+1}.;
    Store the transition tuple $(s_t, \hat{a}_t, r_t, s_{t+1})$ in R.;
  Sample a random mini-batch of $|B|$ transitions $(s_t, \hat{a}_t, r_t, s_{t+1})$ from R.;
  Update the Q-function parameters by using (4.28).;
  Update the execution policy function parameters by using (4.27).
  Update the target-Q function parameters using:

$$\theta^{\text{target}} \leftarrow \tau\theta + (1-\tau)\theta^{\text{target}}$$

DATA AND DISTRIBUTION NETWORK CASE

To validate the effectiveness of our proposed DF-SRL algorithm, we construct an environment based on a CIGRE LV distribution residential sub-network shown in Fig. 4.3. In this network, each node is associated with an aggregator, and the DSO interacts with them to regulate voltage magnitude based on the availability of flexibility at each node. The training data of PVs, plug-in EVs, and typical residential load follows research [142], with a 15-min resolution. The voltage magnitude limit is set as $\overline{v} = 1.05$ $p.u.$ and $\underline{v} = 0.95$ $p.u.$. For the present case study, we assumed that the maximal flexibility provided by the aggregator is 50 kW during the operation [142].

BASELINE METHODS

To evaluate the performance of the proposed DF-SRL algorithm, we conduct a comparative analysis with several DRL benchmark algorithms, including the state-of-the-art DRL algorithms: DDPG, PPO, TD3, and SAC, as well as a centralized model-based approach, i.e., an NLP formulation. The parameters for different DRL algorithms and cases are summarized in Table 4.1. TD3, DDPG and Safe DDPG algorithms are trained with the same hyperparameters as DF-SRL. Specifically, linear safe layer training for safe DDPG follows the default implementation in [85]. All implemented algorithms and their (hyper)parameters are available online[2]. Note that while all the DRL benchmark algorithms can make decisions only using current information and achieve online operation, the so-

[2]Here: https://github.com/ShengrenHou/DF-SRL and here: https://github.com/distributionnetworksTUDelft/DF-SRL

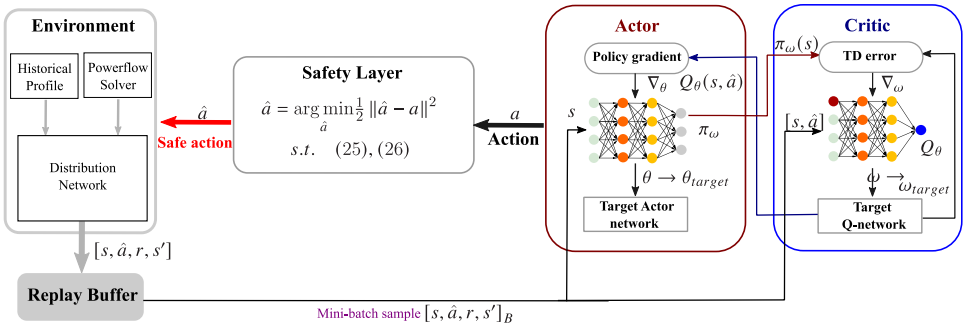


Figure 4.2: Architecture of the proposed DF-SRL algorithm displaying the interaction between the actor and critic networks, the safety layer and the interaction process with the environment (the distribution network simulator).

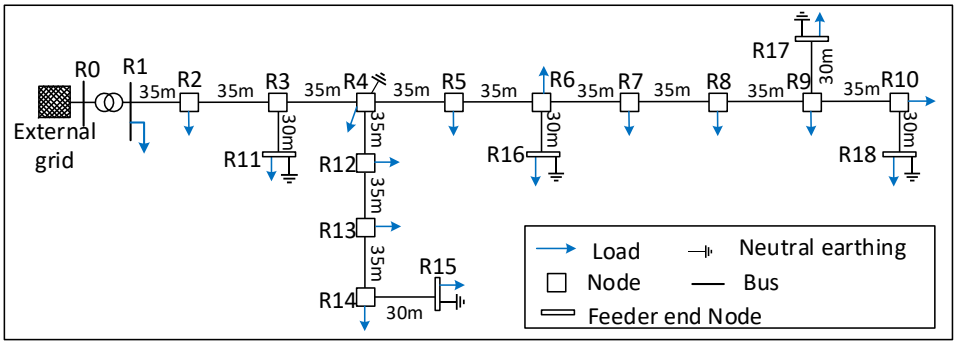


Figure 4.3: Modified CIGRE LV residential network. The aggregator is assumed to coordinate the flexibility each node provides to regulate voltage magnitude.

lution obtained by the NLP formulation requires complete information of the foreseen control period. To train and assess the performance of the DRL benchmark algorithms, we employ validation metrics based on the negative value of total used active power, as denoted in (4.11), and the voltage magnitude violation penalty as specified in (4.13), counted as the cost of the voltage magnitude violation. These metrics effectively gauge the operational efficiency and constraint adherence of each algorithm.

4.4.2. PERFORMANCE ON THE TRAINING SET

Figure 4.4 presents a comparative analysis of the average total reward, the used active power, and the cumulative voltage magnitude violations during the training process for the developed DF-SRL and the benchmark DRL algorithms. Results shown in Fig. 4.4 are obtained as an average of over five algorithm executions. The average total reward increases rapidly during the training, while voltage magnitude violations decrease significantly at the beginning. As depicted in Fig. 4.4(*b*), it is noteworthy that the negative of total used active power for DDPG and TD3 algorithms (with soft penalty) eventually converges around -1.7 MW, while that of SAC and Safe DDPG are -2.4 and -4.3 MW

Table 4.1: Summary - Parameters for DRL algorithms and aggregators

DF-SRL Alg.	$\gamma = 0.995$ Optimizer = Adam Learning rate = $6e-4$ Batch size = 512, Replay Buffer = $4e5$
SAC Alg.	$\gamma = 0.995$ Optimizer = Adam Learning rate = $6e-4$ Batch size = 512, Replay Buffer = $4e5$ Entropy=fixed
PPO Alg.	$\gamma = 0.995$ Optimizer = Adam Learning rate = $6e-4$ Batch size = 4096
Reward	$\sigma = 400$
Aggregator	$\overline{p}^B = 50kW, \underline{p}^B = -50kW$
Voltage limit	$\overline{v} = 1.05, \underline{v} = 0.95$

respectively. Compared to these benchmarks, the DF-SRL figure is significantly lower, at approximately -0.5 MW. Figure 4.4(c) reveals another stark contrast between DF-SRL and the DRL benchmark algorithms (with soft-penalty) and Safe DDPG. Throughout the training process, the DF-SRL consistently enforced the constraints without any voltage magnitude violations, whereas the DRL benchmark algorithms experienced failures in satisfying the constraints after reaching convergence at 1000 episodes. This disparity can be attributed to the safe layer in DF-SRL, which adjusts unsafe actions during training. In comparison, the DRL benchmark algorithms initially grapple with low-quality actions due to the random initialization of the DNN's parameters, leading to many initial violations. Then, based on the guidance of the penalty term of the reward function, the DDPG, TD3, and PPO algorithms decrease the number of voltage magnitude violations to a small value after about 200 episodes. Conversely, the SAC algorithm exhibits slower training efficiency, achieving smaller violation values only at the end of training (1000 episodes). This behavior can be due to the complex exploration policy used by the SAC algorithm. The Safe DDPG algorithm maintained relatively smaller violation values at the beginning compared to other algorithms (e.g., TD3) with a soft penalty. Nevertheless, it fails to enforce violations caused by the poor quality of the safe layer, trained based on the data collected from random policy-environment interaction. Moreover, the unfeasible safe layer also led the action project in the wrong direction, impacting the data quality in the replay buffer, causing a worse performance compared to the standard counterpart (DDPG), as is shown in Fig 4.4(b).

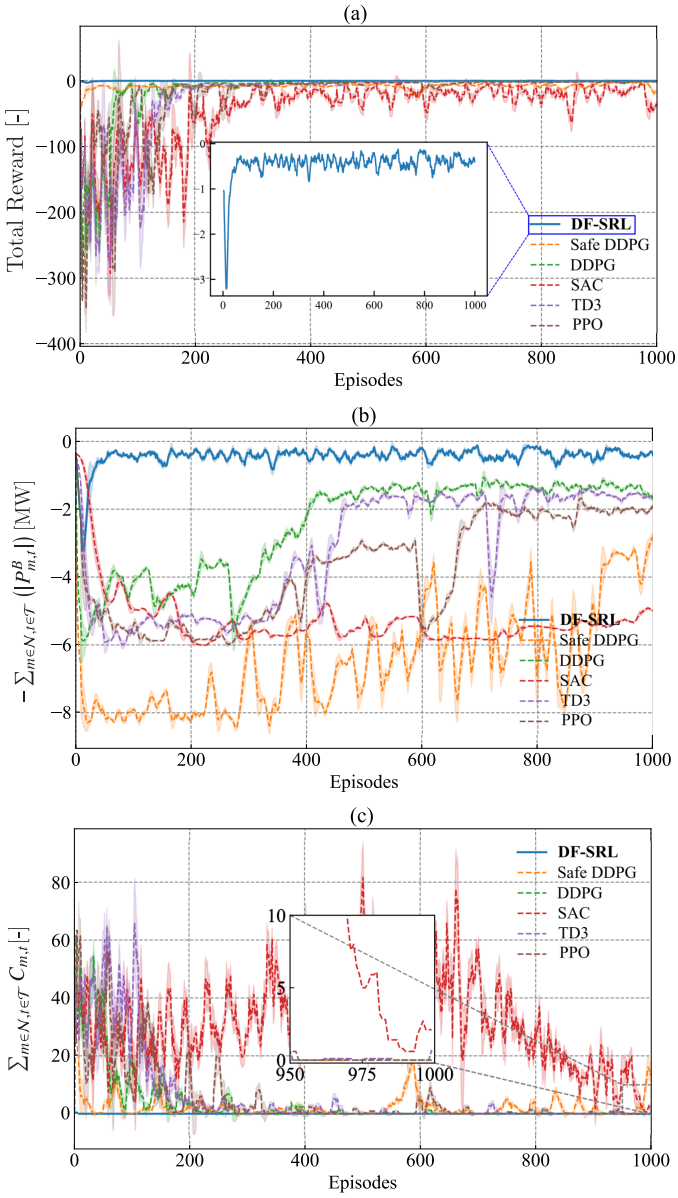


Figure 4.4: (*a*) Average total reward as in (4.17). (*b*) the summation of negative used active power or first term of reward in (4.17). (*c*) Cumulative penalty for voltage magnitude violations or second term of reward in (4.17), all during training.

4.4.3. Performance and Constraint Enforcement Capabilities on Testing Set

Figure 4.5 displays the voltage magnitude results during a typical day in the test dataset. In the specific scenario of nodes 11, 16, 17, and 18 of the network operating under se-

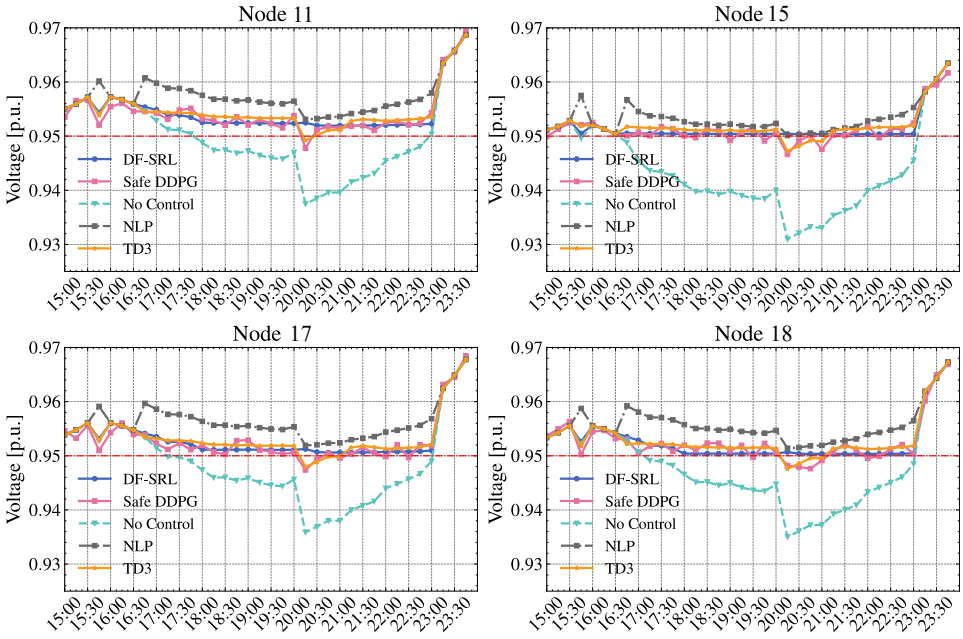


Figure 4.5: Voltage magnitude before and after regulation by the DF-SRL, Safe DDPG and TD3 algorithms, and the NLP formulation. As TD3 performs best among all DRL algorithms, we use the TD3 algorithm as a benchmark.

vere undervoltage during the afternoon and night, the proposed DF-SRL algorithm effectively maintained the voltage magnitude within the technical limits throughout the entire operation period. Notably, the Safe DDPG algorithm failed to maintain the voltage magnitude within the technical limits between 8:00-9:00 pm. This is due to the inherent limitations of the trained linear safe layer, which performs poorly in the distribution network environment with complex dynamics and multiple constraints involved. Similarly, DRL benchmarks, for instance, TD3, trained with a soft penalty, can not provide certified feasibility after convergence. Furthermore, the operational cost associated with DF-SRL's regulation was 0.76 MW, a significant reduction of 17.7% compared to the TD3 and Safe DDPG counterparts. This reduction can be attributed to the high-quality training data provided by the expert knowledge-based safety layer in DF-SRL. Compared with the optimal solution obtained by solving the NLP formulation with a perfect forecast, the DF-SRL demonstrates a modest error rate of 10.6%.

Table 4.2 presents the average total error in operational cost, the average number of voltage magnitude violations (including over and under voltage violations), and the average total computational time for the proposed DF-SRL and (Safe) DRL benchmark algorithms assessed over 30 unseen test days. As illustrated in Table 4.2, the DF-SRL consistently upholds voltage magnitude constraints while achieving a marked reduction in average error relative to the solution obtained by the NLP formulation with perfect forecast. In general, the DF-SRL algorithms perform best of all the algorithms with the

Table 4.2: Performance comparison of different DRL algorithms in an unseen test set of 30 days.

Algorithms	Operation Cost Error [%]	Voltage Magn. Violations [-]	Comp. Time [s]
DF-SRL	11.6 ± 0.0%	0	29±2.4
Safe DDPG	67.1 ± 5.5%	19±2	25±0.7
DDPG	37.2±1.2%	15±4	15.7±0.2
TD3	35.9±1.5%	14±4	15.7±0.2
SAC	56.1±3.4%	23±4	16±0.1
PPO	44.3±1.1%	12±1	15.4±0.6

lowest average error of 11.6%. In contrast, the TD3 algorithm underperforms with an error rate of 35.9%, violating voltage magnitude constraints around 14 time steps. Other DRL algorithms, such as the DDPG, PPO, and SAC algorithms, register higher errors at 37.2%, 44.3%, and 56.1%, respectively. With a trained linear safe layer, the Safe DDPG algorithm fails to enforce voltage magnitude constraints while performing worse than the standard DDPG algorithm. This is because the trained safe layer in the Safe DDPG algorithm can not accurately track the relationship between state, action, and multiple constraints. As anticipated, due to the safety layer's computation, the proposed DF-SRL algorithm requires more computational resources compared with other DRL algorithms. Despite this, the DF-SRL algorithm, as proposed, remains a viable option for real-time operation as it takes less than 29 seconds for one day (96 time-steps) execution.

4.4.4. Sensitivity Analysis

The DF-SRL algorithm capitalizes on the linear relationship between the voltage magnitude and the actions. Nevertheless, the power flow formulation can introduce errors due to the approximation assumptions. The safety layer formulation introduced the slack parameter ϵ to overcome this. Primarily, ϵ should be determined by the upper error boundary for the DistFlow model compared to the actual voltage magnitude. As the final value used for ϵ influences the feasibility and optimality of the actions defined by the DF-SRL algorithm, this section presents an in-depth sensitivity analysis of the slack parameter ϵ.

Figure 4.6 illustrates the convergence performance of the DF-SRL algorithm for different values of the slack parameter ϵ. At $\epsilon = 0.001$, the DF-SRL algorithm performance is markedly diminished after convergence. In this case, the total active power provided by the aggregators is relatively low compared to when ϵ takes the values of 0.002 or 0.005. Additionally, it fails to ensure the feasibility of the decided solutions during training, whereas, with ϵ set at 0.002 or 0.005, the DF-SRL algorithm successfully enforces all operational constraints. In general, a low value of ϵ can make the safe solution of the linear projection model infeasible. Consequently, the resolved safe solution may cause voltage magnitude violations during training, leading to sub-optimal performance after projection. If the DF-SRL algorithm is executed with ϵ set at 0.002 or 0.005, significant performance improvements in optimality and feasibility are observed, as illustrated in Fig 4.6(a, b). Furthermore, the optimality score experienced a modest increase, at around 5%, when ϵ was reduced from 0.005 to 0.002. This can be attributed to the fact that a higher ϵ constrains the solution space in the action projection model, sub-

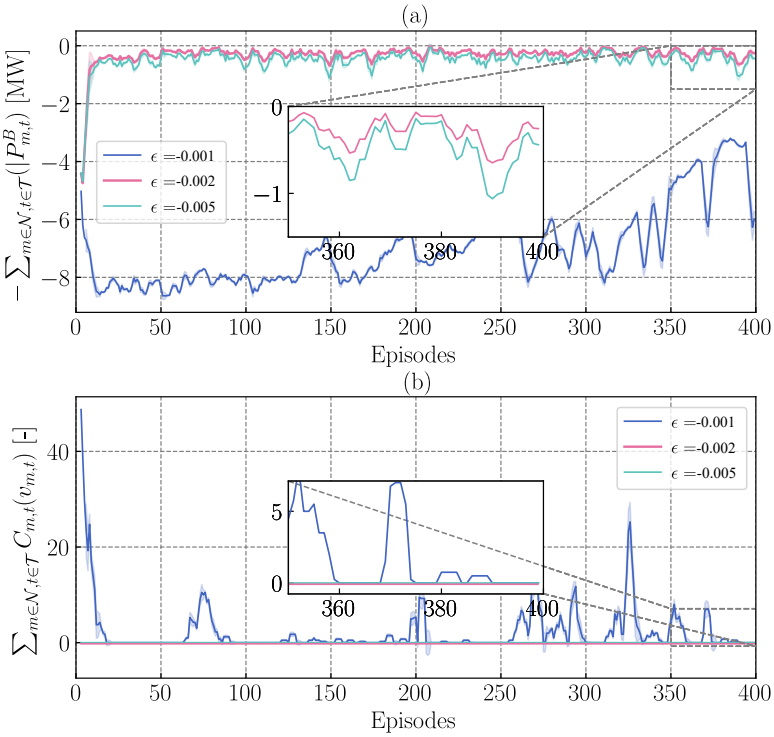


Figure 4.6: (*a*) Total flexible active power, and (*b*) number of voltage magnitude violations; of the proposed DF-SRL algorithms with $\epsilon = (0.001, 0.002, 0.005)$, respectively.

sequently affecting the solution quality during training. The calibration of the slack parameter ϵ is intrinsically linked to the linear error inherent in the safe layer, which is pivotal for the DF-SRL algorithm's efficacy. As such, ϵ is not a Lagrangian multiplier associated with a dynamic constraint penalty, but rather a static safety buffer calibrated based on empirical voltage approximation errors. This calibration ensures that the relaxations provided by ϵ comprehensively cover the linearization errors, thus maintaining the integrity of the safety layer across varying operational scenarios.

In the following section, we conduct a detailed scalability analysis to further explore the range of errors induced by the linearization process, providing a quantitative foundation to refine the selection of ϵ across different network sizes [140].

4.5. SCALABILITY ANALYSIS

The scalability of the proposed DF-SRL algorithm is fundamentally determined by the effectiveness of the DistFlow linearization process. This linearization approximation is essential for mapping the actions from the DRL to safe operational domains. Substantial linearization errors can cause inaccuracies within the safety layer, misguiding action projection, compromising policy iterations, and ultimately degrading the algorithm's overall efficacy.

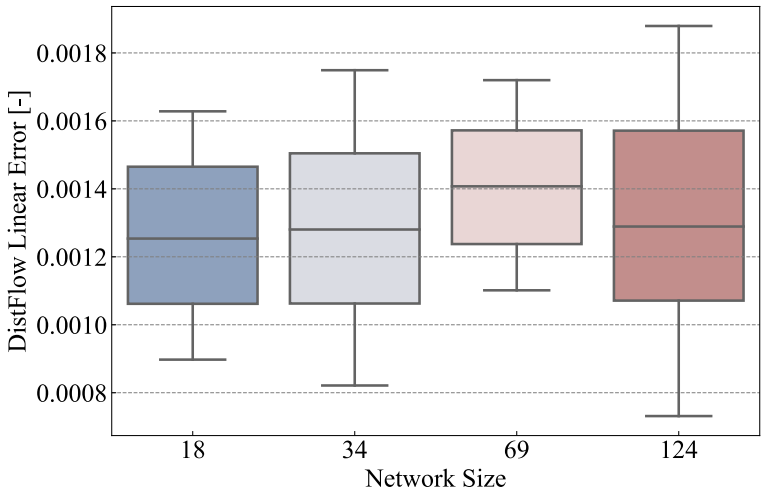


Figure 4.7: Voltage magnitude error of DistFlow on 18, 34, 69, 124 nodes distribution networks. We collect voltage magnitudes from all nodes within networks of 18, 34, 69, and 124 nodes and calculate the deviations between the DistFlow approximations and actual voltage magnitudes in one year's data.

Figure 4.7 presents the observed voltage magnitude errors for different network sizes. The voltage magnitude error in the 18-node distribution network ranged from 0.00089 to 0.00163. In the 34-node network, errors ranged from 0.00082 to 0.00175. The 69-node network experienced an error range of 0.0011 to 0.00172, and the 124-node network saw variability from 0.00073 to 0.00188. Although the largest network exhibited a broader range of error, the maximum error did not exceed 0.002, suggesting that setting an error threshold of $\epsilon = 0.002$ effectively accommodates the inaccuracies induced by the linearization across all tested networks. The results demonstrate the robustness of the DistFlow model, which forms a solid foundation for the safety layer, facilitating its application across diverse distribution network configurations. This generalizability ensures that with precise data on the network's parameters and topology, the safety layer can be tailored to maintain its accuracy and relevance, regardless of the specific characteristics of the network.

5

SAFE IMITATION LEARNING-BASED OPTIMAL ENERGY STORAGE SYSTEMS DISPATCH IN DISTRIBUTION NETWORKS

The integration of distributed energy resources (DER) has escalated the challenge of voltage magnitude regulation in distribution networks. Traditional model-based approaches, which rely on complex sequential mathematical formulations, struggle to meet real-time operational demands. Deep reinforcement learning (DRL) offers a promising alternative by enabling offline training with distribution network simulators, followed by real-time execution. However, DRL algorithms tend to converge to local optima due to limited exploration efficiency. Additionally, DRL algorithms can not enforce voltage magnitude constraints, leading to potential operational violations when implemented in the distribution network operation. This study addresses these challenges by proposing a novel safe imitation reinforcement learning (IRL) framework that combines IRL and a designed safety layer, aiming to optimize the operation of Energy Storage Systems (ESSs) in active distribution networks. The proposed safe IRL framework comprises two phases: offline training and online execution. During the offline phase, optimal state-action pairs are collected using an NLP solver, guiding the IRL policy iteration. In the online phase, the trained IRL policy's decisions are adjusted by the safety layer to maintain safety and constraint compliance. Simulation results demonstrate the efficacy of Safe IRL in balancing operational efficiency and safety, eliminating voltage violations, and maintaining low operation cost errors across various network sizes, while meeting real-time execution requirements.

Parts of this chapter have been submitted to IEEE Journal of Modern Power Systems and Clean Energy with the title: *Safe Imitation Learning-based Optimal Energy Storage Systems Dispatch in Distribution Networks.*

5.1. INTRODUCTION

The penetration of renewable energies has pressed emerging challenges to distribution network operators (DSOs) due to the lag in distribution network upgrades, particularly evident in the Netherlands, where the severity of voltage magnitude problems has escalated [143]. This bottleneck in infrastructure modernization has significantly promoted energy investors to deploy energy storage systems (ESSs) into distribution networks, offering a viable pathway to mitigate voltage magnitude instabilities and enhance the resilience of the distribution network [144]. In this context, optimizing ESSs dispatch is crucial to ensure voltage regulation while also aiming to minimize operational costs amidst the constraints of an aging network [145].

However, fluctuating prices, varying electricity demands, and uncertainty in renewable generation bring significant challenges in defining the dynamic and sequential optimal operation decisions. Traditional model-based approaches, which rely on predefined forecasts or complex probability functions to manage uncertainties, often struggle with real-time decision-making [146]. As these methods require extensive computational resources, they can be inefficient in adapting to the fast-paced and variable nature of the optimal ESSs dispatch problem [147].

Deep Reinforcement Learning (DRL) emerges as a promising alternative to traditional model-based approaches, offering a model-free solution that excels in fast-paced, sequential decision-making scenarios [148]. DRL has been successfully applied in diverse fields such as game playing, robotics control, and industrial systems, where it transforms operational sequences into Markov Decision Processes (MDPs) [149]. In the context of energy systems tasks, DRL has demonstrated the potential to optimize complex tasks, such as voltage control [150] and energy management [151], by enabling the DRL algorithms to learn directly from interactions with the built energy system simulator. This capability allows DRL to handle the complexities and uncertainties inherent in distribution networks more effectively [152]. One of the primary challenges associated with DRL algorithms is low exploration efficiency. The agent requires substantial time to learn due to the need for extensive exploration of the action space [153]. This inefficiency is particularly problematic in scenarios with high-dimensional action spaces, such as controlling multiple ESSs in a distribution network [32]. This low exploration efficiency consequently leads DRL algorithms to converge prematurely to suboptimal solutions, as fully exploring all possible actions becomes increasingly difficult. For instance, previous research [32] has shown that DRL algorithms often focus on leveraging only a single ESS that is highly sensitive to voltage magnitude fluctuations, while neglecting the potential flexibility offered by other ESSs. This behavior results in suboptimal performance and prevents the system from fully utilizing the flexibility of multiple ESSs [154].

Imitation Learning (IL) offers a complementary approach that can enhance the data efficiency of DRL algorithms [155]. IL is a strategy where the learning agent aims to mimic the behavior of an expert by learning from optimal state-action pairs [156]. In the context of ESSs dispatch, expert decisions can be derived from solving daily scenarios using commercial solvers, which derive optimal state-action pairs under various scenarios. These pairs provide a high-quality dataset that the RL agent can use to learn desired behaviors without needing to engage in inefficient online exploration [157]. By incorporating IL, the learning process of DRL algorithms is significantly accelerated, as

the RL agent starts with a base of optimal actions in different states, thereby reducing the exploration space and focusing on refining strategies that have already proven to be effective. For instance, [158] integrated expert demonstrations into the training phase of DRL for real-time dispatch of generation units. Results showed DRL algorithms can achieve faster convergence and improve 2.2% performance compared to the model-based solution in real-time dispatch tasks. The work in [159] applied a Mixed-Integer Linear Programming (MILP)-based IL approach to Heating, Ventilation, and Air Conditioning (HVAC) control. By using IL, a control policy can be trained by imitating the optimal MILP-based decisions, enabling efficient real-time HVAC control without the need for solving complex optimization problems in real-time. In [160], IL is leveraged to accelerate DRL algorithms training for building HVAC control. Results demonstrated DRL algorithms could achieve better control efficiency and effectiveness in managing building HVAC systems. In [161], an IL-based approach is proposed for online optimal power scheduling of microgrids. The IL-based controller can rapidly adjust power scheduling in real-time, ensuring optimal operation of microgrids under varying conditions by learning from optimal scheduling policies derived from offline optimization models.

Previous studies have shown that IL or an offline trained IL followed by online DRL fine-tuning can improve the training efficiency and the performance of dispatch policies. However, this combination presents several challenges. First, purely imitation learning-based approaches are highly sensitive to the training dataset, leading to poor generalizability and potentially suboptimal behavior in scenarios that were not part of the training data [155]. Second, although online fine-tuning can mitigate this problem, it may also cause a performance collapse due to the state-action distribution shift, where the DRL agent's exploration leads to actions and states that deviate significantly from those seen during the imitation learning phase [162]. Third, both of these previous approaches struggle to guarantee the feasibility of the decisions or enforce operational constraints, as they do not explicitly account for feasibility during the imitation learning process [163]. In light of these challenges, our contributions are threefold:

- We introduce a framework that combines the strengths of DRL algorithms and IL to enhance the training efficiency and dispatch performance of trained algorithms. Moreover, the framework can rigorously enforce operational constraints in distribution networks during the dispatch. This innovative approach addresses the limitations previously identified in these areas.

- During the offline training phase, we employ a dual-gradient strategy utilizing both the IL policy and the critic network. This approach stabilizes the training process and expedites learning, effectively overcoming standard DRL algorithms' computational and exploration challenges.

- To guarantee the feasibility of dispatch decisions, the safe layer proposed in our previous paper [118] is extended to the framework during the online operation. This layer filters out unsafe actions, redirecting them into safer alternatives, thus ensuring the operational feasibility of decisions in scenarios not covered by expert data.

5.2. Mathematical Formulation

The optimal scheduling of ESSs within a distribution network is formulated as a non-linear programming (NLP) problem, given by (5.1)–(5.7). The objective function in (5.1) aims to minimize the total operational cost over the time horizon $\mathcal{T}$, which includes the costs of importing power from the main grid, dictated by day-ahead market prices ρ_t in EUR/kWh.

$$\min_{P^B_{m,t},\forall m\in\mathcal{B},\forall t\in\mathcal{T}} \left\{ \sum_{t\in\mathcal{T}} \left[\rho_t \sum_{m\in\mathcal{N}} \left(P^D_{m,t} + P^B_{m,t} - P^{PV}_{m,t}\right)\Delta t \right] \right\}. \tag{5.1}$$

Subject to:

$$\sum_{nm\in\mathcal{L}} P_{nm,t} - \sum_{mn\in\mathcal{L}} (P_{mn,t} + R_{mn} I^2_{mn,t}) + P^B_{m,t} + P^{PV}_{m,t} + P^S_{m,t} = P^D_{m,t} \quad \forall m \in \mathcal{N}, \forall t \in \mathcal{T} \tag{5.2}$$

$$\sum_{nm\in\mathcal{L}} Q_{nm,t} - \sum_{mn\in\mathcal{L}} (Q_{mn,t} + X_{mn} I^2_{mn,t}) + Q^S_{m,t} = Q^D_{m,t} \quad \forall m \in \mathcal{N}, \forall t \in \mathcal{T} \tag{5.3}$$

$$V^2_{m,t} - V^2_{n,t} = 2(R_{mn}P_{mn,t} + X_{mn}Q_{mn,t}) + (R^2_{mn} + X^2_{mn}) I^2_{mn,t} \quad \forall m,n \in \mathcal{N}, \forall t \in \mathcal{T} \tag{5.4}$$

$$V^2_{m,t} I^2_{mn,t} = P^2_{mn,t} + Q^2_{mn,t} \quad \forall m,n \in \mathcal{N}, \forall t \in \mathcal{T} \tag{5.5}$$

$$SOC^B_{m,t} = SOC^B_{m,t-1} + \begin{cases} \frac{\eta^B_{m,c} P^B_{m,t} \Delta t}{\overline{E}^B_m}, & \text{if } P^B_{m,t} > 0 \\ \frac{P^B_{m,t}\Delta t}{\eta^B_{m,d}\overline{E}^B_m}, & \text{if } P^B_{m,t} < 0 \end{cases} \quad \forall m \in \mathcal{B}, \forall \sqcup \in \mathcal{T} \tag{5.6}$$

$$\underline{SOC}^B_m \le SOC^B_{m,t} \le \overline{SOC}^B_m \quad \forall m \in \mathcal{B}, \forall t \in \mathcal{T} \tag{5.7}$$

$$\underline{P}^B_m \le P^B_{m,t} \le \overline{P}^B_m \quad \forall m \in \mathcal{B}, \forall t \in \mathcal{T} \tag{5.8}$$

$$\underline{V}^2 \le V^2_{m,t} \le \overline{V}^2 \quad \forall m \in \mathcal{N}, \forall t \in \mathcal{T} \tag{5.9}$$

$$0 \le I^2_{mn,t} \le \overline{I}^2_{mn} \quad \forall mn \in \mathcal{L}, \forall t \in \mathcal{T} \tag{5.10}$$

$$P^S_{m,t} = Q^S_{m,t} = 0 \quad \forall m \in \mathcal{N}\setminus\{1\}, \forall t \in \mathcal{T} \tag{5.11}$$

The distribution network is modeled using the power flow formulation shown in (5.2)–(5.5) in terms of the active $P_{mn,t}$ power, reactive power $Q_{mn,t}$ and current magnitude $I_{mn,t}$ of lines, and the voltage magnitude $V_{m,t}$ of nodes. Equation in (5.6) models the dynamics of the ESSs' SOC on the set $\mathcal{B}$, while (5.7) enforces the SOC limits. Hereafter, it is assumed that the ESS $m \in \mathcal{B}$ is connected to node m, thus, $\mathcal{B} \subseteq \mathcal{N}$. Finally, (5.8) enforces the ESSs discharge/charge operation limits, (5.9) and (5.10) enforce the voltage magnitude and line current limits, respectively, while (5.11) enforces that only

one node is connected to the substation. Notice that to solve the above-presented NLP formulation, all long-term operational data (e.g., expected PV generation and consumption) must be collected to properly define the ESSs' dispatch decisions, while the power flow formulation must also be considered to enforce the voltage and current magnitude limits.

5.3. MDP FORMULATION

The above sequential-decision problem can be modeled as a constrained Markov decision process (CMDP), characterized by the tuple $(\mathscr{S},\mathscr{A},\mathscr{P},\mathscr{R},\gamma,\mathscr{C})$. Here, $\mathscr{S}$ denotes the state space which includes observable states of the system, $\mathscr{A}$ represents the action space of possible control actions, $\mathscr{P}$ is the state transition probability capturing system dynamics, $\mathscr{R}$ is the reward function guiding the policy iteration, γ is a discount factor reflecting the importance of future rewards, and $\mathscr{C}$ is a set of constraint functions ensuring operational safety and feasibility. The decision-making follows a policy $\pi(a_t|s_t)$ that selects actions $a_t \in \mathscr{A}$ based on the current state $s_t \in \mathscr{S}$, deriving the system along a trajectory of states, actions, and rewards: $\tau = (\boldsymbol{s}_0, \boldsymbol{a}_0, \boldsymbol{s}_1, \boldsymbol{a}_1, \cdots)$. The selected actions aimed at maximizing a cumulative reward while adhering to system constraints.

The state at time t, denoted s_t, encapsulates the current operational status of the distribution network and it is defined by the vector: $s_t = [P^N_{m,t}, V_{m,t}|_{m\in\mathscr{N}}, \rho_t, SOC^B_{m,t}|_{m\in\mathscr{B}}, t]$, where $P^N_{m,t} = P^D_{m,t} - P^{PV}_{m,t}$ represents the net power at node m, incoperating both consumption $P^D_{m,t}$ and PV generation $P^{PV}_{m,t}$. $V_{m,t}$ is the voltage magnitude at node m. t is used to indicate which step the agent is in during the whole trajectory. $SOC^B_{m,t}|_{m\in\mathscr{B}}$ is the ESS connected to m_{th} node.

The action a_t at time t involves the dispatch decisions for charging or discharging ESSs, represented by $a_t = [P^B_{m,t}|_{m\in\mathscr{B}}]$, where $\mathscr{A}$ is a continuous space reflecting the possible charge/discharge power. The transition to the next state s_{t+1} based on the current state s_t and action a_t is captured by:

$$p(S_{t+1}, R_t|S_t, A_t) = \Pr\{S_{t+1} = s_{t+1}, R_t = r_t \mid S_t = s_t, A_t = a_t\}. \tag{5.12}$$

The transition function $p(\cdot)$ incorporates both the deterministic dynamics of the distribution network and the stochastic nature of demand, PV generation, and market prices.

The reward function is designed to reflect the operational cost, defined negatively as:

$$\mathscr{R}_t(s_t, a_t) = r_t = -\rho_t \left[\sum_{m\in\mathscr{N}} \left(P^D_{m,t} + P^B_{m,t} - P^{PV}_{m,t}\right) \right] \Delta t \tag{5.13}$$

To ensure safety and operational feasibility, several constraints are integrated. ESS charging and discharging must not exceed predefined limits (5.8). Voltage and current magnitudes must comply with network standards (5.9), (5.10). While constraints on actions and SOC are enforced directly in the policy π, network constraints are managed indirectly. To handle this, a penalty term is added to the reward function for violations:

$$r_t = -\rho_t \left[\sum_{m\in\mathscr{N}} \left(P^D_{m,t} + P^B_{m,t} - P^{PV}_{m,t}\right) \right] \Delta t - \sigma \left[\sum_{m\in\mathscr{B}} C_{m,t}(V_{m,t}) \right], \tag{5.14}$$

where ρ is a penalty coefficient, and $C_{m,t}$ is a penalty function for voltage violations, defined to prioritize operational constraints within the learning process.
$C_{m,t}$ in (5.14) can be modeled using different functions (e.g., L_2 functions). Here, as in [59], $C_{m,t}$ is defined as

$$C_{m,t} = \min\left\{0, \left(\frac{\overline{V} - \underline{V}}{2} - \left|V_0 - V_{m,t}\right|\right)\right\}, \forall m \in \mathscr{B}. \tag{5.15}$$

While the CMDP framework supports the integration of operational constraints, directly applying DRL to optimize ESS scheduling in distribution networks introduces significant challenges. DRL algorithms face high computational demands and often achieve suboptimal policy convergence, struggling to consistently adhere to operational constraints in complex ESSs dispatch problems [105]. Furthermore, after training, DRL agents may fail to enforce these constraints reliably, especially in scenarios that were underrepresented during training. To tackle these issues, we introduce a safe imitation learning framework, which is detailed in the subsequent section.

5.4. THE PROPOSED FRAMEWORK

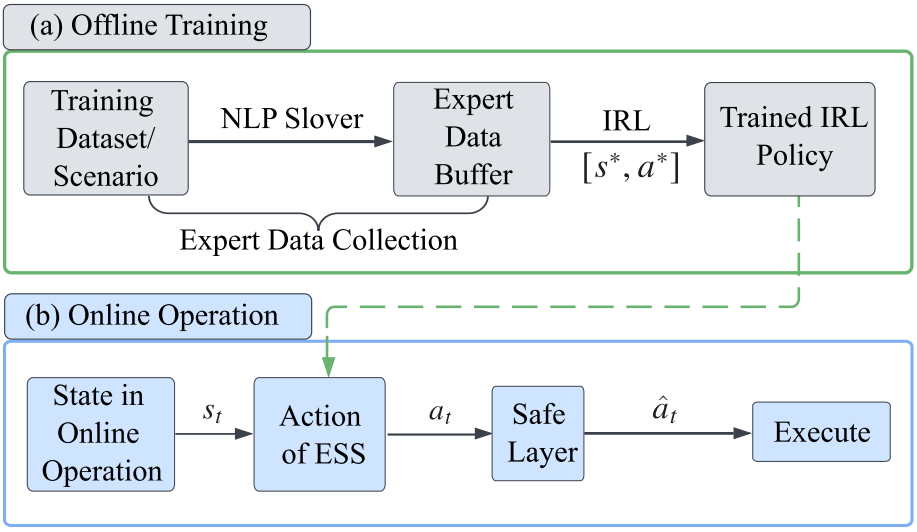


Figure 5.1: Overall workflow of the proposed framework. The framework is composed of offline and online phases. The offline training is performed once, while the online operation is conducted at each time step t.

The proposed framework comprises two main phases: offline training and online execution. Initially, during the offline training phase, an expert policy formulated by an NLP solver collects optimal state-action pairs, or *expert data*. This data is used to guide the IRL policy iteration. In the online execution phase, the trained IRL policy determines charge/discharge decisions based on the current state. These decisions are then adjusted by the safety layer to ensure strict adherence to operational constraints. This

dual-phase approach aims to balance the need for operational efficiency with the essential requirements of safety and constraint compliance.

5.4.1. OFFLINE TRAINING VIA IMITATION LEARNING

EXPERT DEMONSTRATION DATA COLLECTION

The expert demonstration data is crucial for training our IRL framework. Optimal state-action sequences are generated by solving the NLP problem formulated in Section 5.2, capturing a variety of historical scenarios including daily trajectories of renewable generation, load consumption, and price dynamics. This expert policy identifies sequences that minimize operational costs while complying with voltage magnitude constraints, thereby providing a robust dataset for training proposed IRL algorithm.

IMITATION RL ALGORITHMS

Reinforcement Learning (RL) emerges as a preeminent strategy for devising policies under uncertainty. Traditional value-based DRL algorithms, such as DQN [16] fail to address the continuous state and action problems. In contrast, Deep Deterministic Policy Gradient (DDPG) algorithm [23] and it's enhanced counterpart, TD3 [24], are capable of handling continuous actions by simultaneously maintaining a policy (actor) $\pi_\omega(s_t)$, used to sample actions, and a trained Q-function (critic) $Q_\theta(s_t, a_t)$, used to guide the update direction of the policy network. The TD3 algorithm updates the actor-network by

$$\omega \leftarrow \omega + \nabla_\omega \frac{1}{|B|} \sum_{s_t \in B} \left(\min_{i=1,2} \{ Q_{\theta_i}(s_t, \pi_\omega(s_t)) \} \right), \tag{5.16}$$

while the critic update iteration is defined as

$$\min_\theta \sum_{s \in B} \left(r_t + \gamma \min_{i=1,2} \{ Q_{\theta_i^{\text{target}}}(s_{t+1}, \pi_\omega(s_{t+1})) \} - Q_{\theta_i}(s_t, a_t) \right)^2 \tag{5.17}$$

Training the TD3 algorithm to achieve convergence demands extensive interactions between the agents and their environment, a challenge amplified by large state and action spaces. This intensive requirement stems from the necessity for the algorithm to learn from zero. To solve these challenges, the IL approach, specifically behavior cloning (BC), is introduced. BC leverage expert demonstrations to directly map states and actions, thereby significantly enhancing learning efficiency in terms of sample complexity and trading efficiency. Given a dataset of state-action pairs $D^* = (s^*, a^*)$ obtained from expert demonstrations, where s^* represents the states observed by the expert and a^* represents the corresponding actions taken by the expert policy, the goal of BC is to learn a policy $\pi_\omega(s)$ that can generate actions closely approximating the expert's actions for any given state s.

The learning process of policy $\pi_\omega(s)$ involves adjusting ω to minimize the difference between the actions predicted by the policy and the expert actions in the dataset. The parameter update for BC is defined as:

$$\omega^* = \arg\min_\omega \frac{1}{|B|} \sum_{(s^*, a^*) \in D^*} \left| \pi_\omega(s^*) - a^* \right|^2, \tag{5.18}$$

where ω^* represents the optimized policy parameters, B is the batch size, (s^*, a^*) are the state-action pairs from the expert demonstrations, and $\pi_\omega(s)$ is the policy parameterized by ω.

BC aims to train a policy that can accurately replicate the expert's decision-making process across a wide range of states, thereby leveraging the expert's knowledge to achieve efficient learning especially in environments where exploring through trial and error (as in traditional RL approaches) might be inefficient or infeasible. However, the main drawback of BC is that if the learner makes a mistake during execution, it may end up in a state completely distinct from the demonstration dataset, which will consequentially lead to error cascading.

The TD3BC algorithm represents an innovative approach to overcoming the challenges associated with BC, particularly the issue of error cascading when a learner encounters states not covered by the demonstration dataset. TD3BC merges the robustness of DRL with the efficiency of BC for offline training phases.

The TD3BC algorithm integrates the update mechanisms of both TD3 and BC, formulated as:

$$\omega \leftarrow \omega - \alpha \nabla_\omega \left(\lambda_{TD} \frac{1}{|B|} \mathscr{L}_{TD} + \lambda_{BC} \frac{1}{|B|} \mathscr{L}_{BC} \right), \tag{5.19}$$

where: $\mathscr{L}_{TD}$ is the TD loss component, represented by

$$\mathscr{L}_{TD} = \left(r_t + \gamma \min_{i=1,2} Q_{\theta_i^{\text{target}}}(s_{t+1}, \pi_\omega(s_{t+1})) - Q_{\theta_i}(s_t, a_t) \right)^2 \tag{5.20}$$

$\mathscr{L}_{BC}$ is the BC loss component, represented by $|\pi_\omega(s^*) - a^*|^2$, λ_{TD} and λ_{BC} are the weighting coefficients for the TD and BC loss components, respectively, α is the learning rate, B is the batch of transitions sampled from the expert dataset D^*.

TD3BC innovatively combines the gradients from both the conventional TD loss, used in TD3 for updating the policy and value networks, and an expert loss derived from BC. This dual-gradient approach allows the algorithm to not only learn from the expert demonstrations but also refine its policy via interacting with the environment, as in classical RL, thereby addressing the limitations of each approach when used in isolation.

Despite the TD3BC algorithm's ability to enhance performance and accelerate training, it faces a significant limitation during the online execution phase: it cannot inherently enforce constraints. This limitation stems from the fact that the TD3BC algorithm is trained exclusively on demonstration data, which inherently satisfies operational constraints through the resolution of an NLP problem. Consequently, the algorithm, while effective in replicating demonstrated behaviors, lacks an intrinsic understanding of the safety constraints. This gap in awareness can lead to situations where the actions chosen by the TD3BC-trained agent, when faced with scenarios not covered in the training data, diverge from safe operational bounds, potentially causing serious violations of system constraints.

To address this critical issue and ensure the feasibility and safety of actions during online execution, we propose the integration of a linear safe layer on top of the TD3BC algorithm. This safety layer is designed to function as a regulatory mechanism, adjusting the actions suggested by the TD3BC model to ensure they remain within predefined

safety and operational constraints. It acts as a vital check, correcting for the algorithm's lack of direct constraint recognition and ensuring that all actions are compatible with the system's safety requirements.

The next section will explain in detail the formulation and operational mechanism of the safe layer, illustrating its role in maintaining both optimized performance and stringent adherence to operational safety constraints.

5.4.2. ONLINE EXECUTION WITH SAFE LAYER

The safety layer, introduced in our previous work, leverages a linear approximation of power flow equations to project potentially unsafe actions into a safe operational domain. This projection ensures compliance with system constraints during real-time operation.

SAFE LAYER FORMULATION

Building on the linear power flow model detailed in [140], the safety layer adjusts actions based on a simplified relationship between node voltages and power injections. This linear expression can further be used to derive a direct relationship between the action $\boldsymbol{a}$ vector, corresponding to the dispatch decision of the batteries, i.e. $\boldsymbol{a} = [p^B_{1,t}, ...p^B_{m,t}, ...p^B_{|\mathcal{N}|,t}]$, and $\boldsymbol{v^2_m}$, as next

$$\boldsymbol{M}\boldsymbol{v^2} = \boldsymbol{M}v_0^2\mathbf{1}_{|\mathcal{L}|} + 2[\mathrm{D}(\boldsymbol{r_{mn}})\left(\mathbb{I} - \boldsymbol{T}\boldsymbol{F}^T\right)^{-1}\boldsymbol{T}(\boldsymbol{p^N_m} - \boldsymbol{a}) + \mathrm{D}(\boldsymbol{x_{mn}})\left(\mathbb{I} - \boldsymbol{T}\boldsymbol{F}^T\right)^{-1}\boldsymbol{T}\boldsymbol{q^N_m}]. \quad (5.21)$$

$\boldsymbol{v^2_m}$ is vector of squared voltage magnitudes at each node, $\boldsymbol{M}$ denotes Matrix relating node voltages to the power injections in the network, v_0^2 refers squared voltage magnitude at the source node or substation, $\mathbf{1}_{|\mathcal{L}|}$ is vector of ones, the size of which matches the number of lines in the network, $\mathrm{D}(\boldsymbol{r_{mn}})$, $\mathrm{D}(\boldsymbol{x_{mn}})$ are diagonal matrices containing line resistances and reactances, respectively, $\boldsymbol{T}$, $\boldsymbol{F}$ are matrices representing the network topology, specifically the connections between nodes.

The primary focus is on maintaining voltage levels within permissible bounds by modifying the control actions suggested by the RL algorithm. The linear relationship is used to form a mathematical programming problem that finds the closest safe action, minimizing deviations from the initially suggested action while ensuring operational safety:

$$\hat{\boldsymbol{a}} = \arg\min_{\hat{\boldsymbol{a}}} \frac{1}{2}\|\hat{\boldsymbol{a}} - \boldsymbol{a}\|^2. \quad (5.22)$$

Subject to:

$$(\boldsymbol{v^2_m}\mathbf{1}_{|\mathcal{L}|} + 2\boldsymbol{M^{-1}}[\mathrm{D}(\boldsymbol{r_{mn}})\left(\mathbb{I} - \boldsymbol{T}\boldsymbol{F}^T\right)^{-1}\boldsymbol{T}(\boldsymbol{p^N_m} - \boldsymbol{a}) + \mathrm{D}(\boldsymbol{x_{mn}})\left(\mathbb{I} - \boldsymbol{T}\boldsymbol{F}^T\right)^{-1}\boldsymbol{T}\boldsymbol{q^N_m}]) \leq \overline{v}^2 - \epsilon \quad (5.23)$$

$$(\boldsymbol{v^2_m}\mathbf{1}_{|\mathcal{L}|} + 2\boldsymbol{M^{-1}}[\mathrm{D}(\boldsymbol{r_{mn}})\left(\mathbb{I} - \boldsymbol{T}\boldsymbol{F}^T\right)^{-1}\boldsymbol{T}(\boldsymbol{p^N_m} - \boldsymbol{a}) + \mathrm{D}(\boldsymbol{x_{mn}})\left(\mathbb{I} - \boldsymbol{T}\boldsymbol{F}^T\right)^{-1}\boldsymbol{T}\boldsymbol{q^N_m}]) \geq \underline{v}^2 + \epsilon \quad (5.24)$$

Algorithm 5: Online Execution for Safe TD3BC Framework

Initialize safety layer parameters: $\mathrm{D}(\boldsymbol{r_{mn}}), \mathrm{D}(\boldsymbol{x_{mn}}), \boldsymbol{B}, \boldsymbol{T}, \boldsymbol{M}$. Load trained TD3BC model. **for** *each operational timestep* t **do**
  Acquire action a_t from policy $\pi_\omega(s_t)$. **if** *action* a_t *risks constraint violation* **then**
    Adjust a_t to $\hat{a}_t$ using safety layer optimization.
  Implement action $\hat{a}_t$ in the system.

In the above formulation, $\hat{\boldsymbol{a}}$ corresponds to the projected (or safe) action vector. Additionally, due to the error introduced in the linear formulation, a small value ϵ is added to control the relaxation condition of voltage magnitude limits, following the previous research [118].

ONLINE EXECUTION PROCEDURE

The procedure for online execution is illustrated in Algorithm 5. The trained TD3BC model proposes initial actions based on received states. These actions are then adjusted by the safety layer if they risk violating operational constraints. The algorithm ensures that all actions are safe and reliable before implementation in the distribution network.

5.5. SIMULATION RESULTS

To evaluate the performance of the proposed Safe TD3BC algorithm, we conduct a comparative analysis with several representative (safe) DRL benchmark algorithms, including TD3 algorithm, Safe TD3 algorithm and TD3BC algorithm. In addition, a centralized model-based approach, an NLP formulation [142] with perfect forecast information is counted as the global optimality. We first evaluate the performance of the proposed Safe TD3BC algorithm in a 34-node distribution network and then scalability analysis is conducted in diverse sizes of distribution network cases (18-node, 69-node, and 124-node). All these distribution network environments are provided in the open-sourced package [164]. The parameters for different DRL algorithms and cases are summarized in Table 5.1. TD3, Safe TD3 algorithms are trained with the same hyperparameters as safe TD3BC algorithms. The parameters of the implemented safe layer follow our previous research [118]. Note that while all the DRL benchmark algorithms can make decisions only using current information and achieve online operation, the solution obtained by the NLP formulation requires complete information of the foreseen control period. To train and assess the performance of the DRL benchmark algorithms, we employ validation metrics based on the negative value of total used active power, as denoted in (5.13), and the voltage magnitude violation penalty as specified in (5.14), counted as the cost of the voltage magnitude violation. These metrics effectively gauge the operational efficiency and constraint adherence of each algorithm.

5.5.1. PERFORMANCE ON TRAINING SET

Table 5.2 presents the performance and training time of Safe TD3BC and benchmark algorithms applied to a simulated 34-node distribution network. The key metrics evalu-

Table 5.1: Summary - Parameters for DRL algorithms and the environment

TD3, Safe TD3	$\gamma = 0.995$ Optimizer adopts Adam Learning rate is $6e-4$ Batch size is 512 Replay buffer size is $4e5$
TD3BC	$\gamma = 0.995$ Optimizer adopts Adam Learning rate is $6e-4$ Batch size is 512, Replay Buffer is $4e5$ $\lambda_{BC} = 0.5$, $\lambda_{TD} = 0.5$
Safe TD3BC	$\gamma = 0.995$ Optimizer adopts Adam Learning rate is $6e-4$ Batch size is 512, Replay Buffer is $4e5$ $\lambda_{BC} = 0.5$, $\lambda_{TD} = 0.5$
Reward	$\sigma = 400$
ESSs	$\overline{P}^B = 150kW, \underline{P}^B = -150kW,$ $\overline{SOC}^B = 0.8, \underline{SOC}^B = 0.2, \eta_c^B/\eta_d^B = 0.98$

Table 5.2: Performance and training time of algorithms on simulated 34-node distribution network.

Algorithms	Training Time [h]	Converged Reward [-]	Violations [-]
TD3	4.3	1.7±0.1	-0.9±0.1
Safe TD3	17.1	2.4±0.3	-1.6±0.8
TD3BC	0.9	4.9±0.5	-7.7±1.5
Safe TD3BC	0.9	4.5±0.1	0

ated are training time (in hours), converged reward, and violations. A higher converged reward indicates better algorithmic performance, while negative values in the violations column signify undesirable constraint breaches.

The TD3 algorithm achieves a moderate converged reward of 1.7 ± 0.1, with some violations recorded at -0.9 ± 0.1. In contrast, the Safe TD3 algorithm improves the performance with a higher converged reward of 2.4 ± 0.3. However, this improvement comes at the cost of increased violations (-1.6 ± 0.8), suggesting that while the Safe TD3 algorithm optimizes the reward better than the TD3 algorithm, it struggles to adhere to constraints effectively. The TD3BC algorithm emerges as the top performer in terms of the converged reward, achieving the highest value of 4.9 ± 0.5. TD3BC effectively harnesses the potential of offline data, enabling the algorithm to quickly adapt to profitable strategies collected in past operations. Consequently, the TD3BC algorithm shows a marked improvement in performance metrics, capitalizing on the accumulated knowledge embedded in the dataset to optimize actions more efficiently than TD3 and Safe TD3 algorithms. Nevertheless, the TD3BC algorithm caused the most severe violations recorded at -7.7 ± 1.5, indicating that the algorithm does not sufficiently enforce safety constraints. This high performance coupled with significant violations highlights a critical failure

of the TD3BC algorithm to maintain safe operations. In contrast, the Safe TD3BC algorithm presents a more balanced approach, combining high performance with strict constraint enforcement. The Safe TD3BC algorithm achieves a converged reward of 4.5 ± 0.1, slightly lower than TD3BC without any voltage magnitude violations. These results suggest that the Safe TD3BC algorithm effectively optimizes performance while strictly enforcing safety constraints, making it a robust choice for applications requiring high reliability and safety.

5.5.2. DISPATCH DECISION COMPARISION ON TESTING DATASET

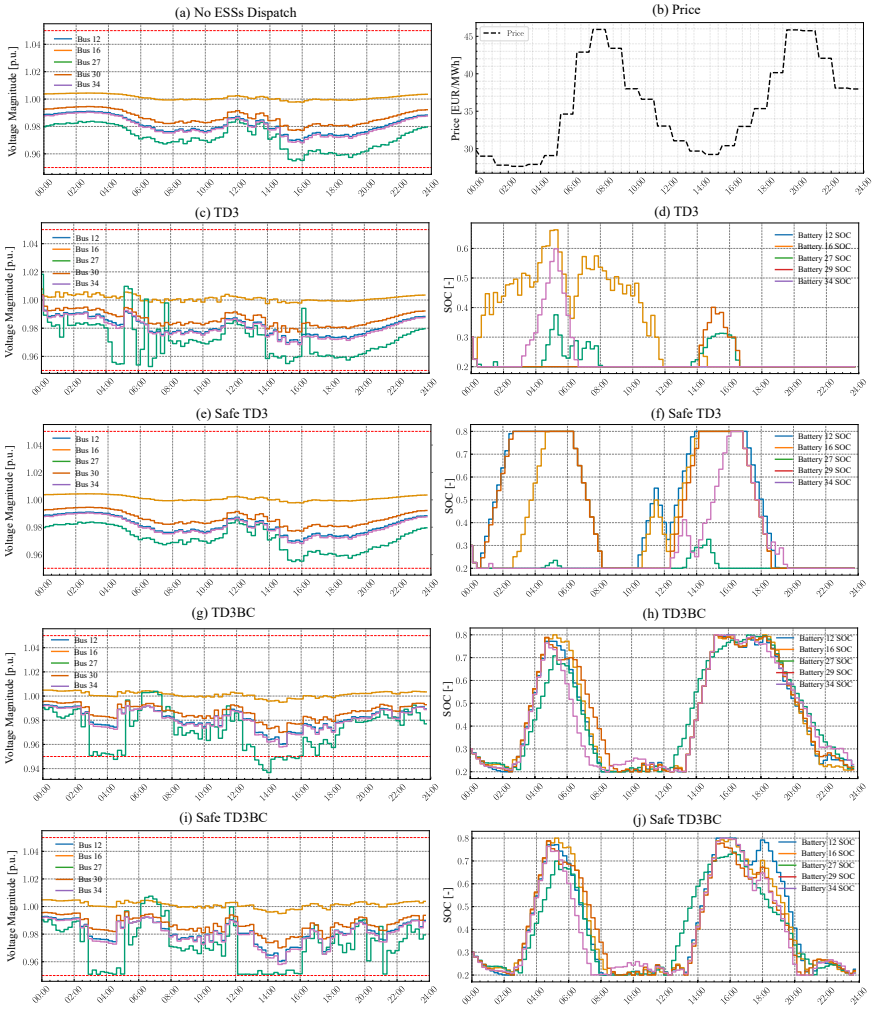


Figure 5.2: (a): Voltage magnitude for nodes in which the ESSs are connected, disregarding their operation. (b): Price in €/MWh. Voltage magnitude ((c), (e) (g)) in which the ESSs are connected and SOC of ESSs ((d), (f), (h)), after executing the dispatch decisions.

Fig. 5.2 displays the voltage magnitude of the nodes in which the ESSs are connected and the SOC of each ESS during a typical day in the test dataset. Results shown in Fig. 5.2 are obtained after using the dispatch decisions provided by the TD3, Safe TD3, TD3BC, and Safe TD3BC algorithms. Fig. 5.2(a) shows the voltage magnitude of the nodes in which the ESSs are connected, but in this case, disregarding their operation (i.e., ESSs are neither charging nor discharging), while Fig. 5.2(b) shows the day-ahead electricity price of that test day.

The TD3 algorithm optimizes ESSs operations by responding to price signals, as shown in Fig. 5.2(d). This enables it to define charging and discharging decisions of ESSs to maximize profit margins. However, the TD3 algorithm does not fully leverage the potential flexibility of all ESSs. For instance, the TD3 algorithm primarily dispatches the ESS connected to Bus 16, largely ignoring the flexibility offered by ESSs connected with other nodes. Additionally, the TD3 algorithm fails to leverage the evening price peaks, indicating convergence to a local optimum. In contrast, the TD3BC algorithm demonstrates a more aggressive strategy, as shown in Fig. 5.2(h). It exploits ESSs flexibility to a greater extent by scheduling ESSs operations aggressively to capitalize on favorable price periods. The TD3BC algorithm maximizes economic gains but at the cost of frequent voltage violations, especially notable during low price periods such as between 02:00 and 04:00, 12:00 and 14:00, causing serious voltage magnitude drops for node 27.

The Safe TD3BC algorithm eliminates the risk of voltage magnitude violations while fully leveraging the flexibility provided by ESSs connected to all nodes. The safety layer actively adjusts the decisions of the TD3BC algorithm, projecting potentially unsafe actions into safe domains. These modifications, which follow the principle of minimizing the Euclidean distance to the original actions, are designed to prevent safety breaches while maintaining the integrity of operational goals. Fig. 5.2(j) shows how the Safe TD3BC algorithm manages the SOCs effectively without causing voltage magnitude violations, as evident from the stable voltage magnitude in Fig. 5.2(i). While the safety layer introduces some trade-offs in terms of reduced economic performance due to necessary adjustments to ensure safety, the overall impact is profoundly positive. Safe TD3BC substantially enhances system reliability, effectively eliminating voltage magnitude violations without significantly compromising on economic benefits.

Fig. 5.3 displays the detailed charge and discharge patterns for the TD3BC and Safe TD3BC algorithms across nodes 12, 16, 27, 30, and 34 between 12:00 and 16:00. The TD3BC algorithm, depicted in the left column of Fig. 5.3, demonstrates a clear strategy of aggressive charging and discharging. For instance, at node 12, The TD3BC algorithm charges the ESSs up to 0.15 MW at 13:15, significantly increasing the SOC. Similar patterns are observed at nodes 16, 27, 30, and 34, where the TD3BC algorithm aims to capitalize on this price period. However, this aggressive strategy leads to serious voltage violations. The aggressive charging caused a significant voltage drop in node 27, leading to voltage violations. In contrast, the Safe TD3BC algorithm, shown in the right column of Fig. 5.3, incorporates a safety layer that modifies the charge and discharge decisions to avoid voltage magnitude violations. The Safe TD3BC algorithm still engages in charging and discharging to maximize operational benefits but also guarantee the feasibility of voltage magnitude constraints. Instead of fully charging at 13:15, the Safe TD3BC algorithm maintains a more moderated charging pattern, ensuring the SOC gradually in-

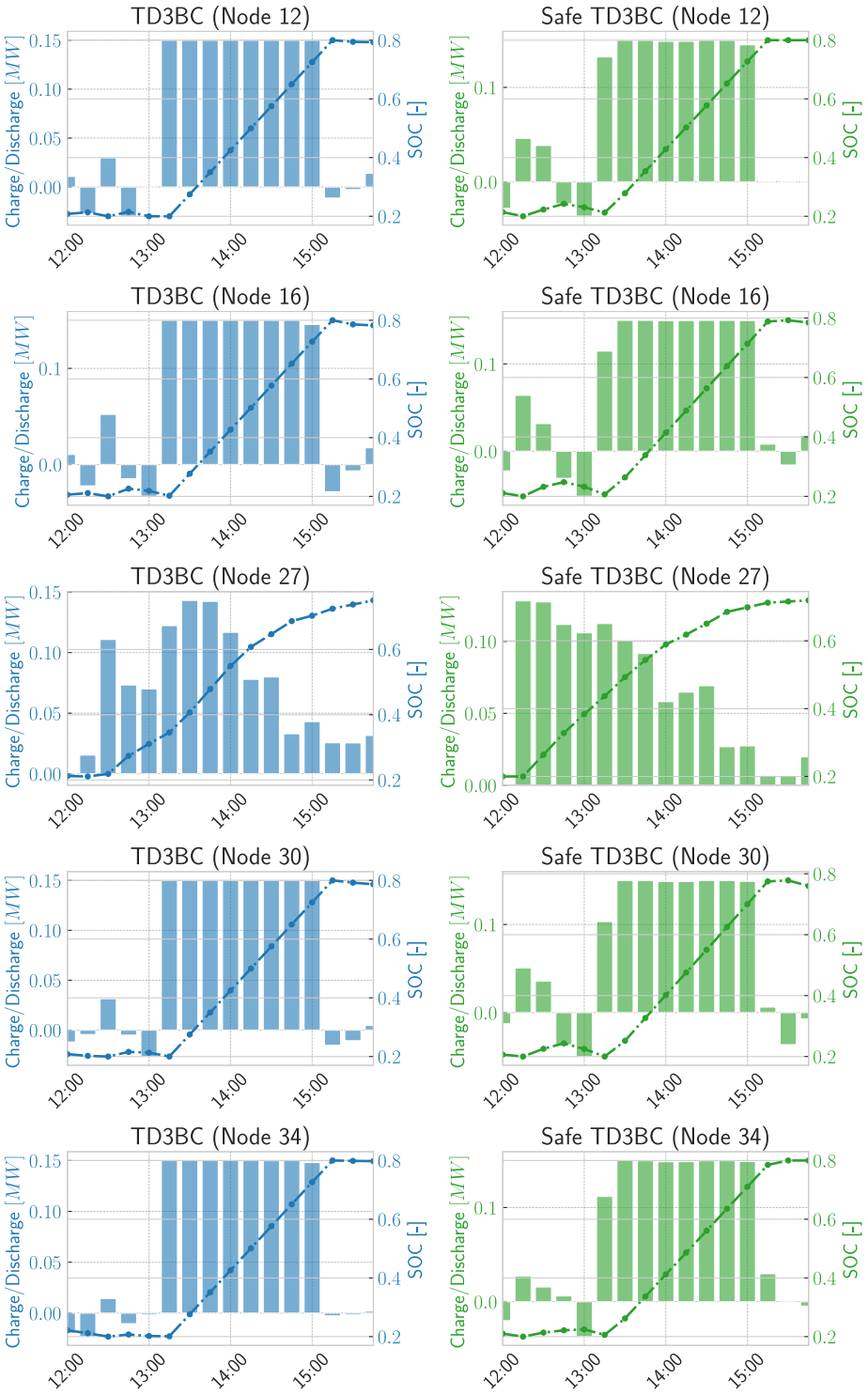


Figure 5.3: ESSs dispatch patterns between 12:00-16:00, conducted by the TD3BC and SafeTD3BC algorithms.

creases without causing voltage magnitude violations. This pattern also modified the decision at 13:00, where the Safe TD3BC algorithm adjusts the charging strategy to prevent the issues observed with the TD3BC algorithm. Across all nodes, the Safe TD3BC algorithm consistently ensures that the SOC increases in a controlled manner. This careful adjustment of charging and discharging schedules highlights the capability of the Safe TD3BC algorithm to balance economic benefits with strict adherence to safety constraints.

5.5.3. SCALABILITY ANALYSIS

Table 5.3: Scalability Analysis of Algorithms on Different Network Sizes

Nodes	Algorithm	Exper Data Collection Time [h]	Training Time [h]	Exec. Time [s]	Operation Cost Error (%)	Voltage Magn. Violations counts [-]
18	TD3	-	4	15±0.1	33.2±1.1	14±2
	Safe TD3	-	12	20±1	15.7±0.8	6±1
	TD3BC	0.5	0.7	15±0.1	3±0.5	45±11
	Safe TD3BC	0.5	0.7	20±0.7	7±0.4	0
34	TD3	-	4	15±0.1	35.9±0.9	19±4
	Safe TD3	-	17	25±1	19.8±1.4	25±7
	TD3BC	1.7	0.9	15±0.1	6±2.5	98±25
	Safe TD3BC	1.7	0.9	22±0.5	10±0.1	0
69	TD3	-	4.9	15±0.1	28.5±0.4	35.1±2
	Safe TD3	-	25	37±2	39.9±5.6	277±87
	TD3BC	2.5	1.5	15±0.1	6.9±0.3	286±35
	Safe TD3BC	2.5	1.5	28±0.5	9.5±0.5	0
124	TD3	-	9	15 ±0.1	49.8±0.4	33±2
	Safe TD3	-	43	75±13	105±10.1	958±109
	TD3BC	9	2.9	15±0.1	11.5±0.7	705±15
	Safe TD3BC	9	2.9	36±1	15.9±2.2	0

Table 5.3 provides a comprehensive overview of the scalability and performance of four different algorithms (TD3, Safe TD3, TD3BC, and Safe TD3BC) across various network sizes (18, 34, 69, 124 nodes). TD3BC and Safe TD3BC algorithms show better performance compared to TD3 and Safe TD3 algorithms, primarily due to their use of expert data. For smaller networks (18 nodes), the TD3BC algorithm achieves an operation cost error of 3 ± 0.5%, significantly lower than the performance of the TD3 algorithm, 33.2 ± 1.1%. However, the TD3BC algorithm fails to enforce safety constraints, resulting in 45 ± 11 violations for the 18-node network. As network size increases, operation cost error of the TD3BC algorithm rises to 11.5 ± 0.7% for the 124-node network, along with a substantial increase in violations (705 ± 15).

In contrast, the Safe TD3BC algorithm consistently maintains low operation cost errors and zero violations across all network sizes. For instance, the Safe TD3BC algorithm has an operation cost error of 7 ± 0.4% for 18 nodes network and 15.9 ± 2.2% for 124 nodes network, without any voltage magnitude violations. This demonstrates the ability of the Safe TD3BC algorithm to balance performance and safety effectively.

TD3 and Safe TD3 algorithms, although not requiring expert data, struggle with constraint enforcement. The TD3 algorithm shows a high number of violations across all network sizes, with 14 ± 2 violations for 18 nodes network and 33 ± 2 for 124 nodes net-

work. The Safe TD3 algorithm performs worse than the TD3 algorithm in larger networks, showing 277 ± 87 violations for 69 nodes and 958 ± 109 for 124 nodes. This indicates that Safe TD3 is not effective in enforcing safety constraints of larger networks.

All algorithms meet real-time requirements, with execution times remaining relatively stable across different network sizes. TD3 and TD3BC algorithms maintain execution times of approximately 15 seconds, while Safe TD3 and Safe TD3BC algorithms have slightly higher execution times due to the additional computations required for enforcing safety constraints. For example, the Safe TD3 algorithm requires 75 ± 13 seconds for a 124-node network, while the Safe TD3BC algorithm requires 36 ± 1 seconds. The lower execution time of the Safe TD3BC algorithm compared to the Safe TD3 algorithm can be attributed to two factors: fewer activation of the safe layer in Safe TD3BC and easier projection of actions, as most actions in Safe TD3BC lie within the boundary.

The preparation of expert data can be time-consuming, as it involves repeatedly solving large-scale optimization problems. This is an offline process and does not impact real-time performance. TD3BC and Safe TD3BC algorithms require less than 3 hours to collect expert data for smaller networks, but this increases to 9 hours for the 124-node network. Training times vary significantly across algorithms and network sizes. Safe TD3 consistently requires more training time than TD3 due to the computational effort involved in ensuring safety constraints. For instance, the Safe TD3 algorithm requires 43 hours to train on a 124-node network, compared to 9 hours for TD3 algorithm. TD3BC and Safe TD3BC algorithms have shorter training times, with the Safe TD3BC algorithm maintaining a training time of 2.9 hours even for the largest network. The extended training time for the Safe TD3 algorithm is primarily due to the frequent activation of the safe layer during environment interactions, which consumes substantial computational resources.

5.6. DISCUSSION

DRL algorithms are designed to optimize decision-making based on the rewards obtained through interactions with the environment. A critical component of their learning process is the exploration of the action space to discover strategies that maximize long-term rewards. However, our findings suggest that standard DRL algorithms often struggle with efficient exploration, particularly in complex operational contexts such as the dispatch of ESSs in distribution networks. One of the key issues observed is that the TD3 algorithm tends to fully dispatch the ESS connected to the node experiencing voltage magnitude issues, while neglecting the dispatch of other ESSs. This behavior leads the algorithm to converge to local optima, rather than exploring more globally optimal strategies. The underlying reason is that DRL algorithms inherently lack mechanisms to sufficiently diversify their exploration, especially in environments characterized by high-dimensional continuous action spaces or intricate reward structures. As a result, once the algorithm identifies a reasonably effective solution, it tends to exploit this solution excessively, foregoing further exploration of potentially superior alternatives [163].

Moreover, while introducing a soft penalty component into the reward structure to enforce operational constraints can help mitigate unsafe actions, it often comes at the cost of overall performance. DRL algorithms must be sensitive to these penalties to avoid violating constraints, which inadvertently shifts the learning focus toward avoiding dan-

gerous actions rather than improving overall performance. This heightened sensitivity to penalties amplifies the significance of actions that frequently lead to violations, causing the algorithm to overemphasize the avoidance of those specific actions. Consequently, the DRL agent may ignore other aspects of the action space that could contribute to better performance, ultimately leading to premature convergence to a local optimum. While the soft penalty approach increases safety, it does so by reducing the algorithm's ability to explore and optimize across other dimensions of the action space, thereby limiting the potential for achieving higher performance.

The TD3BC algorithm integrates BC to accelerate the learning process and improve the action quality by guiding the policy towards historically expert actions. This method effectively harnesses the potential of offline data, enabling the algorithm to quickly adapt to profitable strategies collected in past operations. Consequently, the TD3BC algorithm shows a marked improvement in performance metrics, capitalizing on the accumulated knowledge embedded in the dataset to optimize actions more efficiently than its standard counterpart. Nevertheless, the TD3BC algorithm introduces significant risks related to safety compliance, primarily due to its unawareness of the safety constraints. The expert training dataset cannot encompass all possible real-world scenarios, and when the real-world conditions deviate from the training scenarios, the model may fail to recognize or avoid actions that could lead to operational hazards, such as voltage magnitude violations. This limitation highlights a critical weakness in the TD3BC algorithm: while it can improve performance, it cannot ensure safety under unforeseen conditions.

To address this shortcoming, we propose the Safe TD3BC algorithm, which builds upon the strengths of imitation learning while incorporating mechanisms to guarantee safety. The Safe TD3BC framework not only retains the performance improvements of TD3BC by efficiently dispatching all ESSs, but also introduces a layer of safety that ensures compliance with operational constraints, even in scenarios not covered by the training data. By filtering unsafe actions and providing safer alternatives, the Safe TD3BC algorithm significantly enhances both the performance and safety of ESS dispatch, thus overcoming the limitations of the original TD3BC approach.

6

RL-ADN: A High-Performance Deep Reinforcement Learning Environment for Optimal Energy Storage Systems Dispatch

Deep Reinforcement Learning (DRL) presents a promising avenue for optimizing Energy Storage Systems (ESSs) dispatch in distribution networks. This paper introduces RL-ADN, an innovative open-source library specifically designed for solving the optimal ESSs dispatch in active distribution networks. RL-ADN offers unparalleled flexibility in modeling distribution networks, and ESSs, accommodating a wide range of research goals. A standout feature of RL-ADN is its data augmentation module, based on Gaussian Mixture Model and Copula (GMC) functions, which elevates the performance ceiling of DRL agents. Additionally, RL-ADN incorporates the Laurent power flow solver, significantly reducing the computational burden of power flow calculations during training without sacrificing accuracy. The effectiveness of RL-ADN is demonstrated using in different sizes of distribution networks, showing marked performance improvements in the adaptability of DRL algorithms for ESS dispatch tasks. This enhancement is particularly beneficial from the increased diversity of training scenarios. Furthermore, RL-ADN achieves a tenfold increase in computational efficiency during training, making it highly suitable for large-scale network applications. The library sets a new benchmark in DRL-based ESSs dispatch in distribution networks and it is poised to advance DRL applications in distribution network operations significantly. RL-ADN is available at: https://github.com/

Parts of this chapter have been accepted by Energy and AI with the title: *RL-ADN: A High-Performance Deep Reinforcement Learning Environment for Optimal Energy Storage Systems Dispatch in Active Distribution Networks* [164].

ShengrenHou/RL-ADN and *https://github.com/distributionnetworksTUDelft/RL-ADN*.

6.1. INTRODUCTION

6.1.1. MOTIVATION

Energy Storage Systems (ESSs) play a pivotal role in modern distribution networks, offering enhanced flexibility essential for addressing uncertainties brought by Distributed Energy Resources (DERs) integration [165]. Optimizing ESS dispatch strategies is crucial for distribution system operators (DSOs) to fully harness this flexibility [166]. However, the dynamic and sequential nature of optimal operation decisions, responding to fluctuating prices and varying electricity demands, poses a significant challenge. Traditional model-based approaches often struggle with real-time decision-making due to their reliance on predefined forecasts or complex probability functions to manage uncertainties [167]. Deep Reinforcement Learning (DRL) emerges as a potent model-free solution for such fast-paced, sequential decision-making scenarios, with successful applications in diverse fields like game-playing [168], robotics control [169], industry control [170]. Applied to distribution energy systems, DRL transforms these operational challenges into a Markov Decision Process (MDP), exhibiting impressive results in various energy tasks [171, 172]. DRL's strength lies in its adaptability and capability for real-time decision-making, trained in simulators and then applied to real-world scenarios. This necessitates robust and accurate simulation environments to prevent duplication and provide benchmark frameworks for the development of efficient DRL algorithms.

Therefore, we introduce RL-ADN, an open-source library specifically tailored for DRL-based optimal ESSs operation in distribution networks. It meets diverse research needs while providing customization options for research tasks, ensuring both flexibility and standardization.

6.1.2. RELATED WORK

The RL field has grown significantly, thanks in part to open-source universal simulation environments and benchmark frameworks, like GYM for game-playing [168]. However, this trend is less pronounced in energy system research groups. The absence of such resources hampers the development and integration of DRL algorithms in energy system operation areas. Table 6.1 offers a comparative analysis of functionalities in open-sourced energy system environments. Many existing environments address specific challenges but are often too tailored for broader application [30]. For instance, a microgrid environment is developed to test the performance of DRL algorithms in [30]. The task of formulated MDP is to minimize the power unbalance and operational cost by dispatching distributed generators and ESSs. In the research [26], a distribution network environment is open-sourced to facilitate solving active voltage control problems based on multi-agent RL algorithms. AndesGYM [173] developed an environment for frequency control problems in power systems, which leverages the modeling capability of ADNES and Gym environment. The task is set to minimize the deviations of the frequency value in a given time scope. Consequently, these environments do not lend themselves easily to customization or alterations essential for different or broader research objectives. This specificity leads to fragmentation in the research community, as studies operate in isolation without a standardized benchmark or a universally adaptable toolset.

CityLearn [52] provides an environment for simulating DRL algorithms in charge of

operating building energy systems, in either a centralized (single-agent) or decentralized (multi-agent) way. Focusing on exploring the dynamics inside the building, it ignored the grid-level dynamic. GridLearn [174] is further developed to investigate mitigating over-voltages in the distribution network level by demand response in the buildings. Both two packages simplified the original MDP tasks, by discrete continuous decisions into discrete actions and ignoring the power flow calculation in the distribution networks. PowerGridWorld [175] is a framework for researchers to customize multi-agent environments of power networks, which could integrate existing RL libraries like RLLIb and OPEN-AI BASELINES. PowerGridWorld could work in two ways to implement the multi-agent feature: centralized training and distributional execution, distributional training, and execution. In the environment, OPENDSS is used as an interface to execute the power network operation. Gird2OP [176] is developed to support training an intelligent agent to run a transmission network and has served as a benchmark environment for a series of L2RPN competitions. Grid2OP provided the flexibility for grid modifications, observations, and actions. However, both PowerGridWorld and Grid2OP necessitate extensive power flow calculations during offline training, typically a bottleneck in DRL training, since RL agents need to explore the environments to converge, requiring a large amount of interaction. The mentioned electricity network environments are mainly built based on standard iterative methods, i.e., Newton-Raphson method, which is time-consuming, rendering them unsuitable for integration with DRL algorithms training. GYM-ANM [177] is an open-source environment for solving operation problems in distribution networks, with the primary purpose of using RL algorithms to reduce energy loss (including generation curtailment storage, and transmission losses) under the operation violation constraints. GYM-ANM provides flexibility for customizing energy components, research tasks, network topology, etc. Specifically, it uses a customized simplified power flow simulator to encapsulate the dynamics of a distributional network, which can accelerate the training speed of RL agents significantly. However, the limitations of GYM-ANM are also obvious, as the implemented power flow calculation algorithm can not precisely track the dynamic of physical distribution networks, impeding the transition from simulation to reality for the trained RL agents. Therefore, an advanced power flow calculation algorithm remains a significant imperative to avoid being hindered by the extensive computational demands as well as to reflect the dynamics of physical distribution networks accurately.

Moreover, the key to leveraging DRL for optimal dispatch strategies lies in training with diverse historical data, particularly in environments with uncertain renewable generation, load consumption, and price profiles. The broader the training scenarios, the higher performance ceiling of DRL agents [178, 164]. However, collecting diverse data for specific distribution networks remains challenging, limiting the practical integration of DRL algorithms.

6.1.3. CONTRIBUTIONS

This paper presents RL-ADN, an open-source library for DRL-based optimal ESSs dispatch in active distribution networks. RL-ADN accommodates a wide range of research objectives (i.e., different optimization objectives functions such as congestion management and optimal dispatch) while offering unprecedented customization capabilities.

Table 6.1: Summary of literature in environments of distribution network operation. The content of the table strictly aligns with the novelty we include: power flow integration, data augmentation, benchmark optimality, and flexibility assessment.

Work	Research Task	Power Flow Integration	Data Augmentation	Flexibility and Customization Capabilities
[30]	Optimal energy system scheduling	×	×	×
[26]	Voltage regulation	✓	×	×
CityLearn [52]	Building Energy Management	×	×	✓
GridLearn [174]	Building Energy Management	×	×	✓
PowerGridWorld [175]	Power Network Operation	✓	×	✓
Grid2OP [176]	Transmission Network Configuration	✓	×	✓
GYM-ANM [177]	Distribution Network Operation	✓	×	✓
[105]	Microgrid operation	×	×	×
[179]	EV energy management	×	×	×
[180]	Microgrid Control	✓	×	×
[181]	Microgrid operation	✓	×	✓
[182]	Economic dispatch	×	×	×
[183]	Power system emergency control	✓	×	✓
[184]	Voltage Control	✓	×	×
RL-ADN	Optimal ESSs dispatch in distribution network	✓	✓	✓

This flexibility extends to the modeling of distribution network topologies and the integration of various types of ESSs, thereby allowing for the creation of tailored MDPs. RL-ADN incorporates a novel data augmentation module using a Gaussian Mixture Models-Copula (GMC) approach, enhancing the diversity of training scenarios and thereby the performance of DRL algorithms. Additionally, it introduces the Laurent power flow solver, drastically reducing computation time for power flow calculations tenfold, without sacrificing accuracy [185, 31]. RL-ADN also provides four state-of-the-art (SOTA) DRL algorithms and a model-based approach with perfect forecasts as a standard baseline for comparison. In summary, RL-ADN sets a new standard in DRL-based ESS dispatch with its innovative features, flexibility, and efficiency. It paves the way for more effective and accurate DRL applications in energy distribution networks, representing a significant advancement in the field.

6.2. BACKGROUND

6.2.1. OPTIMAL ESS DISPATCH TASKS IN DISTRIBUTION NETWORKS

ESSs dispatch tasks are inherently sequential decision-making problems. The aim is to minimize operational costs while adhering to constraints that ensure the safe and efficient operation of the distribution network. Such constraints might include maintaining specific voltage magnitude and current levels, state of charge (SOC) operation constraints, etc. This involves responding to market prices, network conditions, and renewable stochastic generation. The ESSs dispatch problem is typically cast as optimization problems with a general mathematical optimization formulation defined by (6.1)–(6.3):

Minimize:

$$f(x) \quad \text{where } x \text{ is the decision variable.} \tag{6.1}$$

Subject to:

$$g(x) < y \qquad \text{(Grid-level constraints)} \tag{6.2}$$

$$b(x) < z \qquad \text{(Energy storage system constraints)} \tag{6.3}$$

The objective function $f(x)$ varies based on different tasks, ranging from minimizing operation cost based on dynamic pricing to regulating voltage magnitude or integrating multiple goals [59]. The effective dispatch of ESSs is crucial, considering the uncertainties in renewable generation, load consumption, and price fluctuations. The constraints are categorized into grid-level (6.2) and ESS-level (6.3) based on the specific requirements of the tasks. Some tasks may prioritize network reliability and incorporate more stringent constraints on voltage magnitude and current levels, while others may focus solely on profit maximization. This flexibility in formulation allows for a wide array of approaches, each tailored to meet the specific needs and priorities of different energy optimization tasks.

6.2.2. MDP FORMULATION AND REINFORCEMENT LEARNING

In RL-ADN, these sequential decision-making problems can be reformulated as a MDP, defined by the tuple $(\mathcal{S}, \mathcal{A}, \mathcal{P}, \mathcal{R}, \gamma)$, where $\mathcal{S}$ denotes the state space, $\mathcal{A}$ represents the action space, $\mathcal{P}$ is the state transition probability function, $\mathcal{R}$ signifies the reward function, and γ stands for the discount factor.

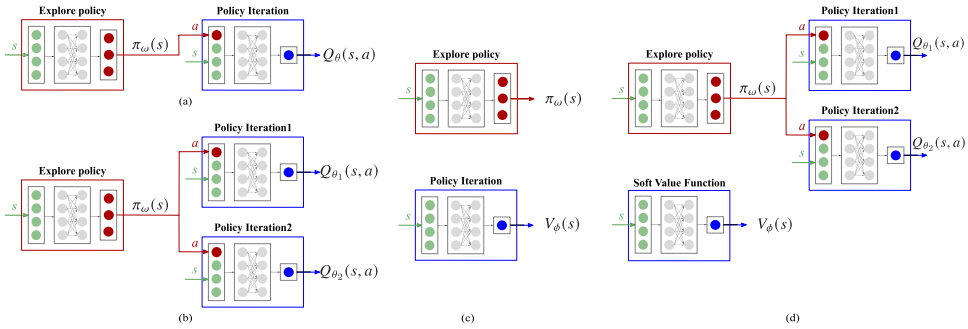


Figure 6.1: Architecture of policy-based DRL algorithms. (a) Deep Deterministic Policy Gradient (DDPG), (b) Twin Delayed DDPG (TD3), (c) Proximal Policy Optimization (PPO), (d) Soft Actor-Critic (SAC).

A policy, $\pi(a_t|s_t)$, determines the selection of action a_t for a given state s_t. The agent's objective is to ascertain a policy that maximizes the expected discounted cumulative return, represented as $J(\pi) = \mathbb{E}_{\tau\sim\pi}\left[\sum_{t=0}^{\mathcal{T}} \gamma^t r_t\right]$, in which $\mathcal{T}$ is the length of the control horizon.

The formulated MDP possesses a continuous action space, making it unsuitable for direct solutions using value-based DRL algorithms [13]. Policy-based DRL algorithms are often employed to address continuous action spaces, as they directly tackle such continuous action domain problems. The architectures of state-of-the-art (SOTA) policy-based DRL algorithms such as DDPG [23], TD3 [24], SAC [77], and PPO [22] are depicted in Fig. 6.1.

- **DDPG and TD3:** Both are deterministic algorithms that maintain a policy for action sampling and Q-networks, $Q_\theta(s_t, a_t)$, to guide policy network updates. Specifically, TD3, as an enhancement of DDPG, incorporates dual Q-networks and employs delayed updates, mitigating the Q-network's overestimation bias inherent in DDPG.

- **PPO:** As an on-policy algorithm, PPO addresses policy optimization challenges in RL. PPO curtails extensive policy updates by adopting a clipped objective function, ensuring minimal deviation of the new policy from the previous one. A value function $V_\phi(s)$ is leveraged to guide the policy iteration. This mechanism circumvents the necessity of learning rate adjustments and achieves superior sample efficiency compared to conventional policy gradient techniques [22].

- **SAC:** SAC is an off-policy actor-critic framework that integrates the maximum entropy reinforcement learning paradigm. By supplementing the typical reward with an entropy component, SAC promotes exploration, thereby achieving a harmonious balance between exploration and exploitation. This algorithm utilizes a soft value function, dual Q-functions, and a policy network. With iterative updates, SAC strives to formulate a stochastic policy that is both optimal and exploratory, ensuring robustness and efficiency across diverse tasks.

Building on the policy gradient theorem, both the policy, $\pi(a_t|s_t)$, and its associated critic networks, $Q_\theta(s_t, a_t)$ or $V_\phi(s)$, can be updated. It is worth noting that the update methods can vary depending on the specific algorithm. A comprehensive discussion of these algorithms is available in [12].

By interacting with the artificial environment, the DRL agent seeks to define the optimal ESSs dispatch in active distribution networks. The two-phase approach, offline training followed by online deployment, equips the agent to address the stochastic nature in optimal ESSs dispatch tasks. In the offline training phase, the DRL agent gleans insights from the interaction and executes self-learning, refining its decision-making. During the subsequent online deployment, it leverages these insights to navigate complexities, ensuring more robust and adaptive solutions. The environment's partially observable nature, often due to communication constraints, necessitates meticulous state selection from the full observation set. Overly complex states will decrease the signal-to-noise ratio, while overly simplistic states could overlook essential dynamics. Both scenarios can undermine the learning efficacy and policy performance. To provide flexibility in designing state spaces, RL-ADN facilitates the easy customization of state spaces, a topic further explored in the subsequent sections.

6.3. RL-ADN Framework

6.3.1. Overview

The architecture of the RL-ADN environment, depicted in Fig. 6.2, consists of three layers: Data Source, Configuration, and Interaction Loop. Primary data feed into Configuration Layer to build DRL environments, integrating components like Data Manager, Distribution Network Simulator, and ESSs Models. These components are integrated into the environment within the Interaction Loop, while a DRL algorithm, chosen to control the agent, is initialized simultaneously[1]. Then, the DRL agent interacts with the environment in search of the optimal policy. The proposed RL-ADN framework's versatility allows for modeling highly tailored tasks, with modifications to components yielding unique MDPs for distinct ESSs dispatch tasks.

6.3.2. Data Source Layer

The Data Source Layer provides primary data for building the framework and training the DRL agent. Data are categorized into time-series data, distribution network data, and ESSs parameter data. Time-series data include load profiles, price profiles, and renewable generation profiles in a standard format. These data are processed by the Data Manager for training or can be selected for further augmentation. Distribution network data comprise node and line data, with nodes specifying slack and PQ bus locations, and lines detailing topology and characteristics like resistance and reactance which are stored in CSV format. This data is crucial for building the distribution network simulator. ESSs parameter data, detailing capacity, charge/discharge limits, and degeneration costs, are used to construct the ESSs model. The framework includes standard 25, 34, 69, and 123 node distribution network data, along with corresponding time-series data and

[1] State-of-the-art policy-based algorithms such as DDPG, SAC, TD3, and PPO are incorporated into the framework.

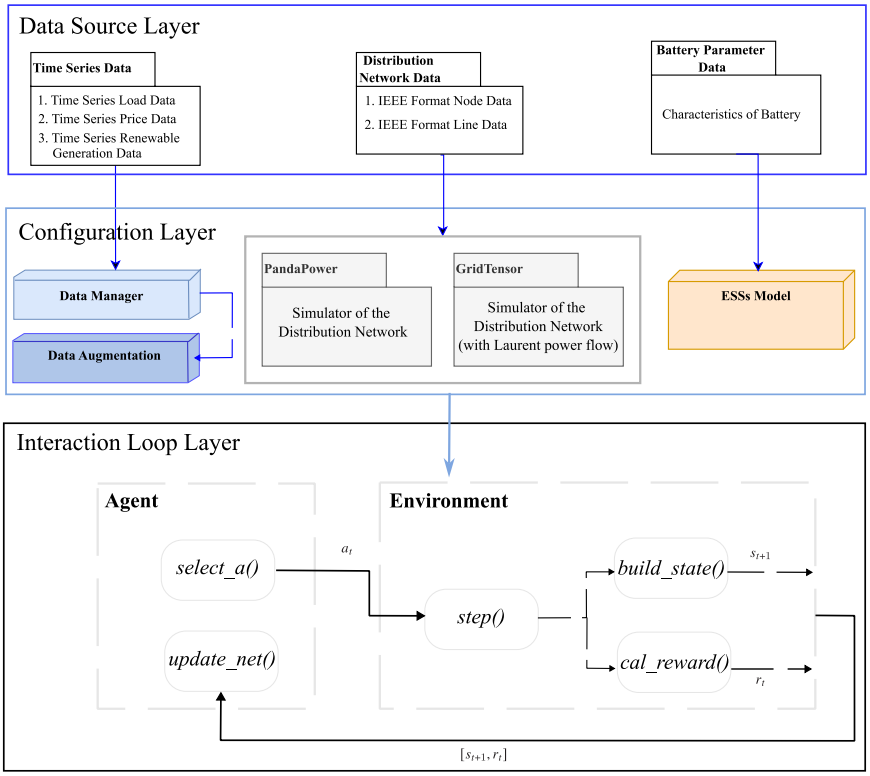


Figure 6.2: Framework of the RL-ADN package. Configuration data for the distribution network and the ESSs are selected from data sources. Subsequently, corresponding time-series data undergo preprocessing. Through Configuration Layer, the environment is constituted of the distribution network, ESSs, and data manager.

ESSs data from previous research [105]. Users can use this data for training or customize their own model following the provided standard format.

6.3.3. CONFIGURATION LAYER

DATA MANAGER

The Data Manager plays a crucial role in managing time-series data, such as active and reactive power demand $(p^D_{i,t}, q^D_{i,t})$, electricity price (ρ_t), and renewable power generation $(p^R_{i,t}, q^R_{i,t})$ for specific epochs $(\mathcal{T}, t \in \mathcal{T})$. Previous research approaches to data management have been case-specific and labor-intensive, adding complexity and potential data quality issues. RL-ADN adopts a streamlined approach, standardizing various data preprocessing tasks, and ensuring data integrity and efficient handling. The workflow of the Data Manager is detailed in Appendix B.1.

Data Augmentation

In RL-ADN, Data Augmentation module plays a pivotal role in enhancing the robustness and generalizability of the trained policy by artificially expanding the diversity of the historical time-series data. With data augmentation, RL-ADN exposes the model to a broader set of scenarios, promoting adaptability and performance in varied and unforeseen situations. The Data Augmentation module is designed to generate synthetic time-series data, capturing the stochastic nature of load in the power system and reflecting realistic operational conditions. The Data Augmentation module interacts with the Data Manager to retrieve the necessary preprocessed data and then applies its augmentation algorithms to produce an augmented dataset. The output is a synthetic yet realistic dataset that reflects the variability and unpredictability inherent in distribution network systems. This enriched dataset is crucial for training RL agents, providing them with a diverse range of scenarios to learn from and ultimately resulting in a more adaptable and robust decision-making policy. The workflow of Data Augmentation module is described in Appendix B.2.

Distribution Network Simulator

For a distribution network, node-set $\mathcal{N}$ and the line set $\mathcal{L}$ define the topology. Each of the node $i \in \mathcal{N}$ and lines $l_{i,j} \in \mathcal{L}$ specify its attributes. A specific subset $\mathcal{B}, \mathcal{B} \subset \mathcal{N}$ describes ESSs connected to the distribution network nodes. Importantly, the number of ESSs delineates the resulting state space $\mathcal{S}$ and action space $\mathcal{A}$.

The main function of the Distribution Network Simulator is to calculate power flow, when a new scenario is fed into the environment, performing as the main part of the state transition function for the formulated MDP task. Based on the provided distribution network configuration data, we offer two modules, `PandaPower` and `GridTensor` to create the Distribution Network Simulator. `PandaPower` provides the traditional iterative methods while `GridTensor` [185] integrates a fast Laurent power flow for calculating the distribution network state presented by the voltage magnitudes, currents and power flowing in the lines.

6.3.4. Interaction Loop Layer

For each time step t in an episode, the agent obtains the current state s_t and determines an action a_t to be executed in the environment. Once a_t is received, the environment will execute `step` function to execute power flow, and update the status of ESSs and the distribution network, which is counted as the consequence of the action at the current time step t. Then, based on these resultant observations, the reward r_t is calculated by the designed reward calculation block. Next, the Data Manager in the environment samples external time-series data of the next time step $t+1$, including demand, renewable energy generation, and price, emulating the stochastic fluctuations of the environment. These external variables are combined with updated internal observations, performing as the resultant transition of the environment.

Users can freely design the `build-state` block, facilitating an in-depth exploration of how different states influence the performance of algorithms on various tasks. In a similar vein, the `cal-reward` block can be tailored according to different optimal tasks. For the convenience of our users, our framework provides a default state pattern and

reward calculation.

6.3.5. MDP DESIGN

STATE SPACE DESIGN

State space design is vital as it directly impacts the efficacy of the agent's learning process. The chosen state space $\mathscr{S}$ should be concise yet descriptive enough to facilitate effective policy learning.

In the RL-ADN framework, the environment collect a comprehensive range of measurements at each timestep t. Using all these measurements to represent the state s_t in the MDP is plausible but fraught with challenges. Such an approach might not be practical in real-world distribution networks due to potential data unavailability. Moreover, by including all measurements, the state space could become noise-prone, making state exploration more intricate and possibly hindering agent performance.

Thus, feature engineering is pivotal in designing state s_t. The RL-ADN framework offers the flexibility to tailor state space. The `get-obs` block fetches available measurements, while the `build-state` block lets users customize states. Generally, the state s_t encompasses both endogenous and exogenous features. Exogenous features capture external dynamics, like uncertainties in renewable energies, consumption, and pricing, within an episode. Meanwhile, endogenous features track internal dynamics governed by distribution network rules and energy component behaviors, e.g., power flow and ESS's SOC update rules. Moreover, some ancillary information, such as the current timestep in a trajectory, has proven crucial in MDP state representation [59].

ACTION SPACE DESIGN

Focusing the optimal ESS dispatch tasks, the action a_t at time t is denoted as $a_t = [p^B_{m,t}|_{m\in\mathscr{B}}]$, symbolizing the charging or discharging directives for the m_{th} ESS connected to node m in the distribution network.

TRANSITION FUNCTION

In a MDP, the transition function encapsulates the dynamics that govern the system's progression from one state to another. The transition mechanism is bifurcated into two essential components. The first is endogenous distribution network and energy component dynamics. These are calculated based on physical laws, i.e., power flow calculation, SOC update rules, rooted in the network's topology, the variations in active and reactive power at different nodes, and the parameter of ESSs models. The second is exogenous variable evolution, which involves modeling the temporal fluctuations in renewable energy generation, market prices, and load demand, leveraging daily historical data. The transition probability function $\mathscr{P}$ is mathematically represented as:

$$p(S_{t+1}, R_t|S_t, A_t) = \Pr\{S_{t+1} = s_{t+1}, R_t = r_t \mid S_t = s_t, A_t = a_t\}. \tag{6.4}$$

Traditionally, constructing a precise mathematical representation of $\mathscr{P}$ has been challenging due to the inherent complexities and uncertainties in both endogenous and exogenous variables. Reinforcement Learning (RL) offers a way around this by learning the ambiguous model through interaction.

REWARD FUNCTION

The reward function serves as a critical component for guiding the agent's learning process. The environment offers a reward signal r_t to the agent, quantifying the quality of each action taken. The design of this reward function is inherently tied to the specific objectives of the task at hand [2]. Our framework incorporates a `cal-reward` block that allows researchers to easily customize the reward signal for various optimal ESS dispatch challenges.

6.3.6. DATA AUGMENTATION MODEL

The RL-ADN framework incorporates Gaussian mixture models (GMM) and Copula functions for data augmentation [186, 187]. The GMM is a probabilistic model that assumes data originates from a blend of multiple Gaussian distributions, each characterized by unique means and covariances. This model can adeptly capture the complex and multimodal nature of time series data in distribution networks, which often exhibit intricate patterns due to fluctuating load demands and renewable energy generation. Complementing the GMM, Copula functions are utilized to encapsulate the time-correlation structure between multiple time-step data in a defined period, independent of their marginal distributions. This dual approach ensures a comprehensive and realistic augmentation of time-series data in distribution network operations. In our framework, three augmentation methods are provided GMM, t-Copula, and Gaussian Copula [178].

The integration of GMM and Copula functions (GMC) in the RL-ADN framework marks a significant advancement in creating robust and reliable environments for training reinforcement learning agents. This approach adeptly handles the complexities and uncertainties inherent in power distribution networks, enhancing the training data's quality and the resulting policies' effectiveness.

6.3.7. LAURENT POWER FLOW

Conventional power flow calculations often rely on iterative methods like the Newton-Raphson algorithm. This becomes a computational bottleneck, especially in the context of training DRL agents, which requires numerous evaluations of power flow. In the proposed framework, we address the computational bottleneck associated with traditional power flow calculations, by incorporating a Laurent power flow algorithm [185]. This efficiency approach is achieved by linearizing the power flow equations using a Laurent series expansion, which simplifies the nodal current calculations in the distribution network. By doing so, we facilitate frequent power flow evaluations necessary for training RL agents, without the computational burden.

The Laurent power flow method we employ considers both constant power and constant impedance loads, integrating the ZIP load model directly into the power flow analysis. This approach allows for the inclusion of various types of loads and renewable energy sources without the need for iterative approximation methods typically used in traditional power flow analysis. As a result, our algorithm achieves rapid convergence and permits a more streamlined and scalable RL training process. The elimination of iterative computation not only expedites the power flow assessment but also enhances the

[2] The default reward functions are presented in Section 4.1.

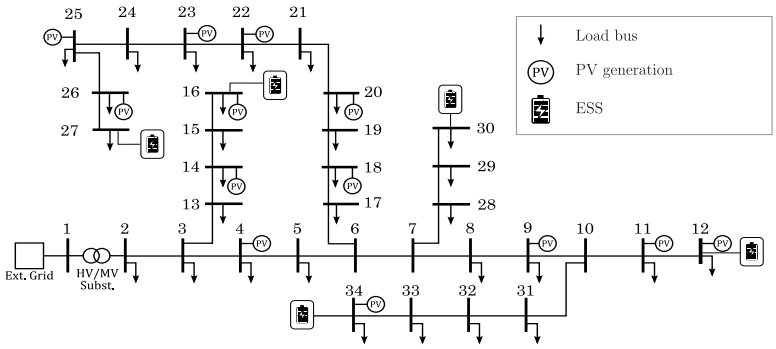


Figure 6.3: Modified IEEE-34 Node bus test system with distributed PV generation and EESs. The ESSs are placed at the end of each feeder to increase the number of voltage magnitude issues experienced.

RL agent's ability to quickly adapt and learn, thereby improving the overall efficiency and effectiveness of the framework.

6.4. BENCHMARK SCHEME AND EXPERIMENTS

6.4.1. OPTIMAL ESSS DISPATCH TASK AND MDPS

RL-ADN framework introduces a foundational optimal ESSs dispatch case while the mathematical formulation of the case is shown in Appendix B.1. This default case aims to minimize the operational costs for DSOs while ensuring compliance with the distribution network and ESSs operation constraints. The template case offers researchers and practitioners a springboard, enabling them to design bespoke benchmarks tailored to unique ESSs dispatch challenges.

In the provided case, a modified 34-node IEEE test distribution network is leveraged to build the Distribution Network Simulator, as illustrated in Fig. 6.3. Strategic placement of the ESSs on nodes 12, 16, 27, 30 and 34 which have over- and under-voltage issues. The objective remains to minimize the operational cost, while upholding voltage magnitude constraints. Consequently, the state, and reward functions are constructed as below: the state s_t is described as $s_t = [P^N_{m,t}|_{m\in\mathcal{N}}, \rho_t, SOC^B_{m,t}|_{m\in\mathcal{B}}]$, incorporating both endogenous and exogenous features. The design of $\mathcal{A}$ adheres to the optimal goal and multiple constraints:

- **Charge and Discharge Bounds:** ESSs have inherent physical limitations. The action a_t is confined within a range, considering these physical constraints.
- **State-of-Charge (SOC) Dependency:** Actions must respect the current SOC of each ESS. The 'step' function ensures this by adjusting the charge/discharge commands based on SOC levels.
- **Voltage Magnitude Regulation:** ESS actions should maintain voltage within predefined limits. Direct enforcement is infeasible; hence, we employ soft constraints via penalty rewards for voltage violations.

Thus, the reward function is defined as the combination of energy arbitrage profits

and the penalty of the voltage magnitude violations in the distribution network. Mathematically, this is expressed as:

$$r_t = \rho_t \left[\sum_{m \in \mathcal{N}} \left(P^B_{m,t} \right) \right] \Delta t - \sigma \left[\sum_{m \in \mathcal{B}} C_{m,t}(V_{m,t}) \right], \tag{6.5}$$

where $C_{m,t}$ is constraint violation functions [167]:

$$C_{m,t} = \min \left\{ 0, \left(\frac{\overline{V} - \underline{V}}{2} - \left| V_0 - V_{m,t} \right| \right) \right\}, \forall m \in \mathcal{B}. \tag{6.6}$$

where σ is a trade-off parameter between energy arbitrage and voltage stability.

6.4.2. BENCH-MARKING APPROACH

To assess performance, we formulate the optimal ESS dispatch problem as a model-based optimization problem, with ESS dispatch decisions as the primary variables. Historical data — including renewable generation, load consumption, and market prices — are treated as perfect forecasts and inputted into the optimization model. Solving this model yields a globally optimal solution, serving as a benchmark for evaluating DRL-derived strategies. Following previous research [30], we can assess the efficiency of DRL algorithms by defining performance bound:

$$\text{Performance Bound} = \frac{C_{DRL} - C_{opt}}{C_{opt}} \tag{6.7}$$

Where C_{DRL} is the operational cost of the dispatch strategy derived from DRL agents, while C_{opt} is that derived from the global optimal solution. The closer the DRL decisions align with this benchmark, the higher the efficacy of the RL agents. We incorporate SOTA DRL algorithms capable of handling continuous action spaces, such as DDPG, PPO, SAC, and TD3, as our benchmark DRL algorithms.

Following prior research [87], our simulation dataset comprises electricity market prices from the Netherlands, augmented with consumption and PV generation data at a 15-minute resolution. Hyperparameter settings for the utilized DRL algorithms are detailed in Table 6.2. We compare the performance of these DRL algorithms against global optimal solutions obtained by formulating Nonlinear Programming (NLP) problems, solved using the Pyomo package [84].

6.5. RESULTS

6.5.1. PERFORMANCE OF DRL ALGORITHMS ON TEMPLATE OPTIMAL DISPATCH TASK

Fig. 6.4 displays the average total reward, operational cost, and the number of voltage magnitude violations during the training process for DDPG, SAC, TD3, and PPO algorithms. Results shown in Fig. 6.4 are obtained as an average of over five random seeds. The average total reward increases rapidly during the training, while simultaneously, the number of voltage magnitude violations decreases. This is a typical training trajectory of DRL algorithms solving optimal dispatch formulated MDP tasks, especially for those

Table 6.2: Summary - Parameters for DRL algorithms and the MDP

PPO Alg.	$\gamma = 0.995$ Optimizer = Adam Learning rate=$6e-4$ Batch size = 4096 GAE parameter(λ) = 0.99
DDPG, TD3 Alg.	$\gamma = 0.995$ Optimizer = Adam Learning rate=$6e-4$ Batch size = 512 Replay buffer size = $4e5$
SAC Alg.	$\gamma = 0.995$ Optimizer = Adam Learning rate=$6e-4$ Batch size = 512 Entropy=auto
Reward	$\sigma = 400$
ESSs	$\overline{p}^B = 50kW, \underline{p}^B = -50kW,$ $\overline{SOC}^B = 0.8, \underline{SOC}^B = 0.2,$
Voltage limit	$\overline{v} = 1.05, \underline{v} = 0.95$

using penalty as a reward. At the beginning of the training process, the DNN's parameters are randomly initialized, and as a consequence, the actions defined usually are random discharge/charge decisions, causing a high number of voltage magnitude violations, thus introducing a huge magnitude penalty term in reward (6.5). Such a reward acts as an indicator to guide updating the DNN's parameters, resulting in higher quality actions, primarily learning to reduce voltage magnitude violations. Then, after reducing the violations, DRL algorithms learn to improve the actions toward increasing and minimizing the operational costs. All these DRL algorithms converged at around 1000 episodes. The total reward of these algorithms converged at 7.5 ± 0.02. Notice that even converged, the operational cost shown in Fig. 6.4(b) will not remain the same because the different daily load and price profiles are sampled during the training. After the last training episode, the penalty voltage magnitude violation penalty for these DRL algorithms was reduced to a value of no more than 1 as is shown in Fig. 6.4(c). This result shows that DRL algorithms can effectively learn from interactions, reducing the number of voltage magnitude violations while minimizing the operational costs by learning to dispatch the ESSs correctly.

Fig. 6.5 shows the dispatch decisions and SOC changes of the ESS, connected to node 16 in a typical daily operation. These decisions are defined by DDPG, TD3, PPO, and SAC, as well as the global optimality benchmark solution provided by solving the NLP formulation considering the perfect forecast. Decisions provided by all DRL algorithms all responded to the dynamic prices during the day. On this day, PPO and SAC perform better than DDPG and TD3. Between 1:00-5:00, when the electricity price is low, PPO and SAC dispatch the ESS in charging mode, which is similar to the decisions from the NLP

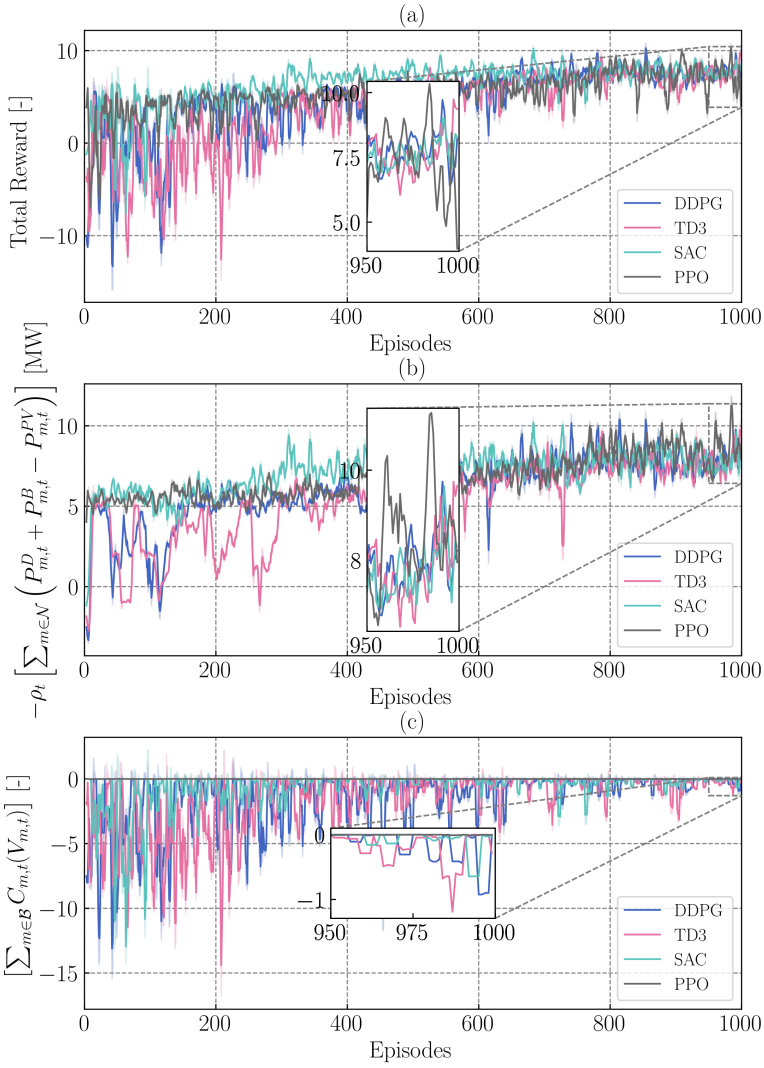


Figure 6.4: (*a*) Average total reward as in (6.5). (*b*) Operational cost or first term of reward in (6.5). (*c*) Cumulative penalty for voltage magnitude violations or second term of reward in (6.5), all during training.

solver. However, DDPG and TD3 fail to learn to act efficiently with the low prices in these timeslots. During the afternoon, all DRL algorithms charge ESSs between low-price slots while discharging between high-price time slots (see Fig. 6.5(*b*) and (*c*)). However, Both DRL algorithms fail to capture the price fluctuations perfectly, compared to the decisions from NLP with full observation of the future. For instance, DDPG performs best among all DRL algorithms between 14:00 and 20:00 but fails to capture the price fluctuations well in the morning. PPO generally performs well during the whole day's operation but defines conservative decisions from 6:00 to 14:00.

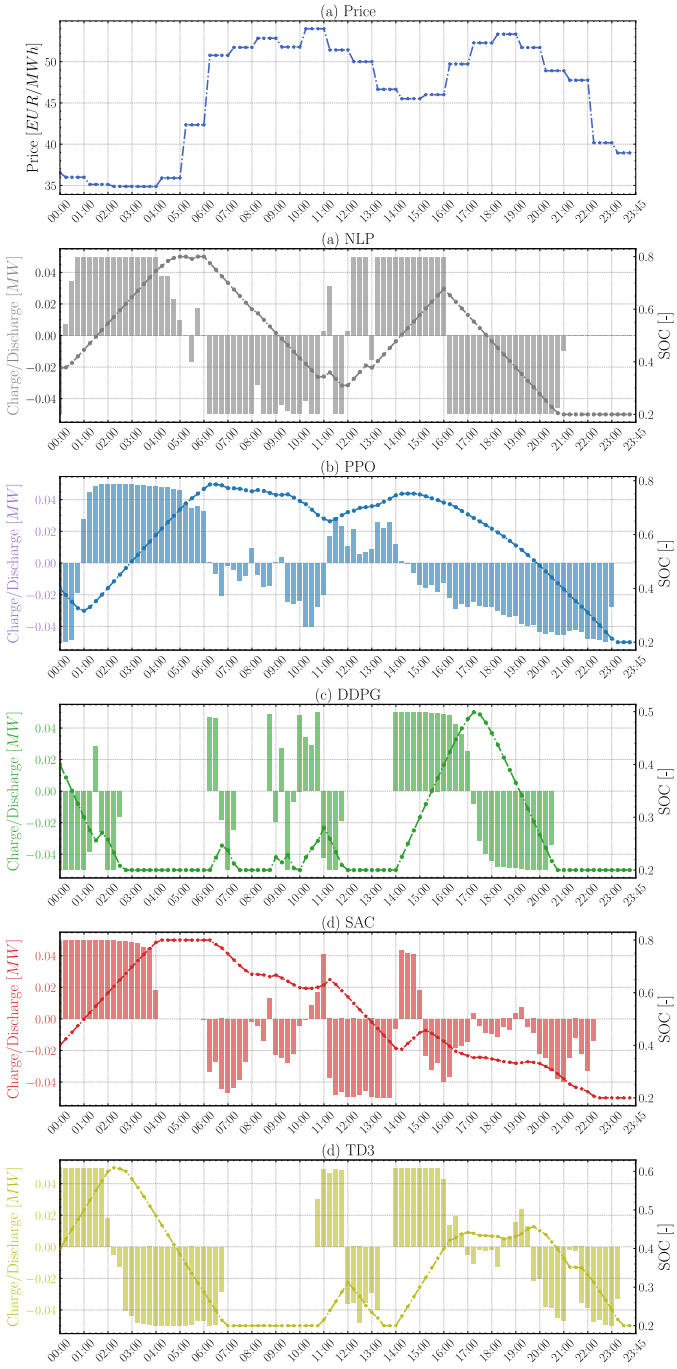


Figure 6.5: Dispatch decisions obtained by DRL algorithms and NLP for the ESS connected to node 16

Compared to the solution provided by NLP, all DRL algorithms converge to a local optimum after training in the current historical dataset. This performance can be caused by the limited scenarios in the training dataset, which hinder the implication of DRL algorithms in the realistic optimal dispatch operation. In the next section, we show how the performance of DRL algorithms is significantly influenced by using the data augmentation model incorporated in the RL-ADN framework.

6.5.2. Impacts of Data Augmentation on Performance of DRL algorithms

The original data and results generated by the GMC model are depicted in Fig. 6.6. The GMC model captured the original patterns of peaks and valleys and diverse scenarios between different nodes in the testing distribution network. For instance, in the original data, the daily consumption profiles at around noon are diverse, where some nodes equipped with ESSs have negative load consumption (discharged), while others show peaks of daily consumption. The developed GMC model replicates such diversity.

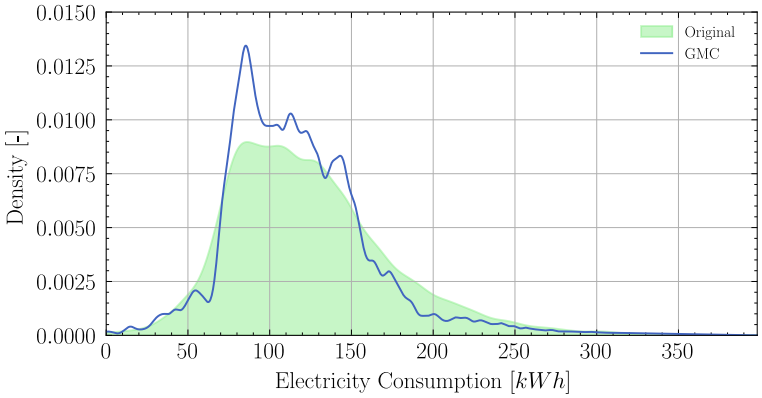


Figure 6.6: Distribution of the original and generated data.

Fig. 6.7 shows the original and generated data distribution shape. Both original and generated data have a long tail distribution. The shape of the GMC augmentation model's distribution matches the original data's shape. Therefore, the generated data profiles can enhance the scenario diversity without losing the original distribution and time correlation in the original dataset.

Table 6.3 presents the average reward, voltage magnitude violation penalty, and performance bounds for DRL algorithms on a separate 30-day test dataset. These algorithms, trained on primary datasets of 1 month, 3 months, and 1 year, were further augmented to 1 year and 5 years to examine the effects of data augmentation within the RL-ADN framework. Consistency in training parameters was maintained across 1000 episodes, and the results include 95% confidence intervals.

Initially, the performance of DRL algorithms using 1-month data was suboptimal. For example, the PPO algorithm's highest performance bound was below 70% (69.1%). However, post-augmentation, there was a significant improvement: PPO's performance

Table 6.3: Mean and 95% confidence bounds for reward, violation penalty and performance bound.

Primary Dataset	**Augmented Dataset**	**Reward** [-]	**Violation Penalty** [-]	**Performance bound** [%]
One month	No augmentation	DDPG (3.40±0.86)	DDPG (0.0±0.0)	DDPG (51.1±6.7)
		PPO (5.91±0.91)	PPO (-0.002±0.001)	PPO (69.1±4.8)
		SAC (4.825±0.62)	SAC (0.0±0.0)	SAC (62.5±4.1)
		TD3 (3.49±0.88)	TD3 (0.0±0.0)	TD3 (52.4±7.0)
	augment 1 year	DDPG (9.55±0.88)	DDPG (-1.05±-0.77)	DDPG (82.8±1.1)
		PPO (11.625±0.92)	PPO (-0.039±-0.01)	PPO (84.0±1.0)
		SAC (9.95±0.63)	SAC (-0.25±-0.01)	SAC (83.4±0.5)
		TD3 (10.565±0.91)	TD3 (-0.09±-0.01)	TD3 (83.9±0.9)
	augment 5 year	DDPG (7.37±0.92)	DDPG (-0.32±-0.22)	DDPG (76.35±4.31)
		PPO (12.59±0.88)	(PPO-2.10±-0.69)	PPO (85.9±1.07)
		SAC (8.25±0.69)	SAC (-0.18±-0.09)	SAC (79.58±1.93)
		TD3 (8.02±0.91)	TD3 (-0.96±-0.41)	TD3 (78.82±2.67)
Three Month	No augmentation	DDPG (8.54±0.99)	DDPG (0.0±0.0)	DDPG (80.4±2.3)
		PPO (6.73±0.97)	PPO (0.0±0.0)	PPO (73.5±4.2)
		SAC (6.92±0.72)	SAC (0.0±0.0)	SAC (74.3±3.1)
		TD3 (8.60±0.92)	TD3 (0.0±0.0)	TD3 (80.6±2.1)
	augment 1 year	DDPG (9.38±0.99)	DDPG (0.0±0.0)	DDPG (82.5±1.4)
		PPO (9.68±0.94)	PPO (0.0±0.0)	PPO (83.0±1.0)
		SAC (7.78±0.55)	SAC (0.0±0.0)	SAC (78.0±1.9)
		TD3 (9.24±0.92)	TD3 (0.0±0.0)	TD3 (82.2±1.4)
	augment 5 year	DDPG (9.24±0.89)	DDPG (0.0±0.0)	DDPG (82.19±1.4)
		PPO (8.72±0.97)	PPO (0.0±0.0)	PPO (81.01±3.1)
		SAC (6.02±0.71)	SAC (0.0±0.0)	SAC (69.71±3.75)
		TD3 (8.45±0.95)	TD3 (0.0±0.0)	TD3 (80.20±3.32)
One year	No augmentation	DDPG (7.061±0.93)	DDPG (-0.01±0.0)	DDPG (75.0±3.7)
		PPO (8.173±1.02)	PPO (0.0±0.0)	PPO (79.3±2.8)
		SAC (7.302±0.84)	SAC (0.0±0.0)	SAC (76.1±3.2)
		TD3 (7.325±1.03)	TD3 (0.0±0.0)	TD3 (76.2±3.8)
	augment 5 year	DDPG (7.58±0.79)	DDPG (0.0±0.0)	DDPG (77.20±2.76)
		PPO (8.91±0.87)	PPO (0.0±0.0)	PPO (81.44±1.71)
		SAC (8.47±0.86)	SAC (0.0±0.0)	SAC (80.26±2.12)
		TD3 (7.99±0.99)	TD3 (0.0±0.0)	TD3 (78.72±2.90)

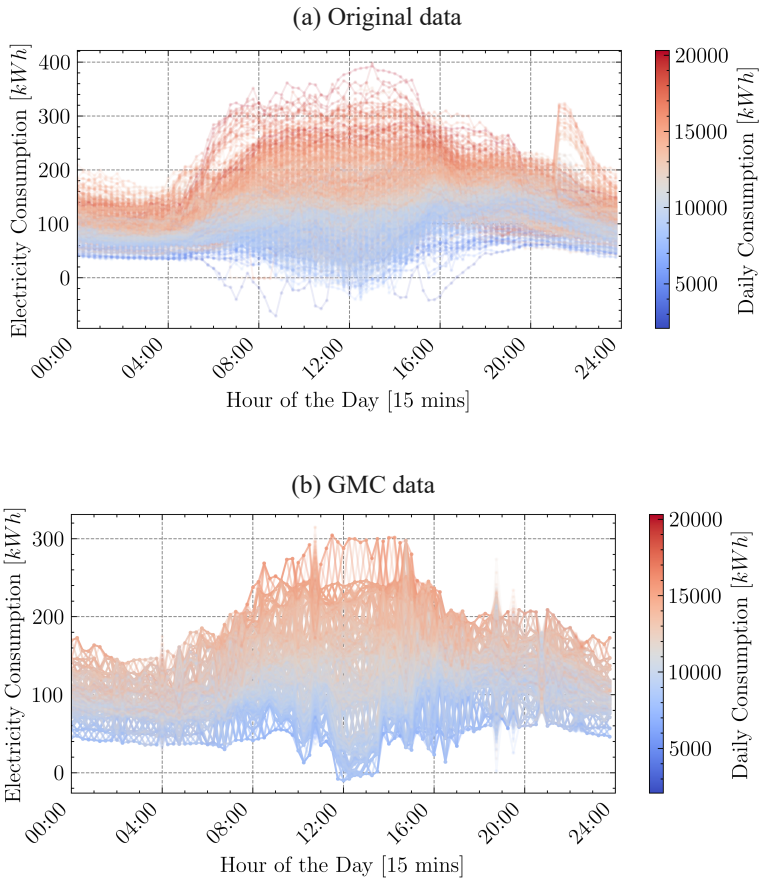


Figure 6.7: Original and GMC generated load profiles. The color of the profiles corresponds to the sum of daily consumption.

increased to 84.0% and 85.9% with 1-year and 5-year data augmentation, respectively. When trained on 3-month primary data, DRL algorithms demonstrated good performance, which was further enhanced with data augmentation. For instance, TD3 improved from 80.6% to 82.2% with 1-year augmentation. Similarly, algorithms trained on one-year primary data showed good performance with minimal test set violations, and augmentation yielded incremental performance gains, as seen with PPO's increase from 79.3% to 81.44%. These results underscore the significance of data augmentation in enhancing the adaptation of DRL algorithms to varied market conditions, particularly for algorithms like DDPG and TD3. In scenarios with limited original datasets, the data augmentation module in the RL-ADN framework can substantially raise the performance ceiling of DRL algorithms.

However, a concerning observation was the increase in voltage magnitude violations in the 1-month data set trained algorithms post-augmentation, particularly notable with

the 5-year augmentation. This could be attributed to the augmented data increasing scenario diversity but not altering the data distribution, as illustrated in Fig. 6.7. In such cases, while DRL algorithms perform better within the existing data distribution, they may incur violations in extreme scenarios not encountered during training. Notably, algorithms trained on more diverse datasets (three-month and one-year) exhibited better control over voltage violations. This is likely because these datasets encompassed the extreme scenarios present in the test sets. Yet, when comparing performance, algorithms trained on the one-year dataset displayed a lower performance ceiling than those trained on the three-month dataset. This suggests that while the one-year data provides a more diverse training environment, leading to potentially better generalization, it also presents a slower learning curve due to its complexity.

Generally, results indicate that in scenarios with limited original datasets, the data augmentation module in the RL-ADN framework can substantially raise the performance ceiling of DRL algorithms. Moreover, the distribution of data and the diversity of scenarios significantly impact the performance of DRL algorithms. Scenario diversity raises the performance ceiling, while data distribution affects the training difficulty and performance in extreme scenarios. While augmentation improves overall performance, it introduces complexities like increased violation penalties, especially when the primary dataset has a limited data distribution.

6.5.3. ENHANCEMENT OF COMPUTATION EFFICIENCY

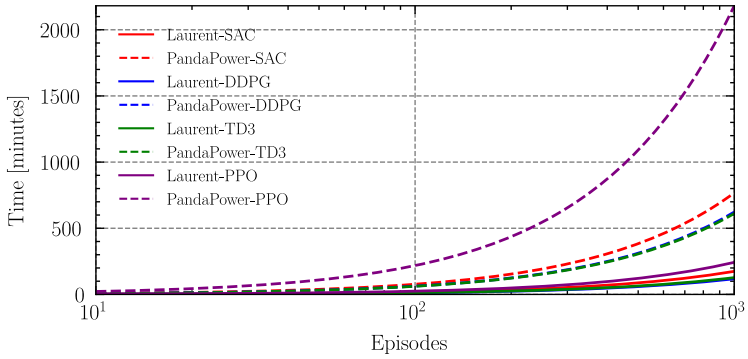


Figure 6.8: Training time for DRL algorithms with Laurent power flow and Panda power. The 34-node distribution network is used as a benchmark.

The performance comparison between Laurent power flow and PandaPower power flow was conducted across multiple scale distribution networks with node sizes: 25, 34, 69, and 123. The summarized results in Table 6.4 indicate a distinct computational advantage for the Laurent power flow method over PandaPower. First, Laurent power flow consistently maintained its efficiency, taking less than 1 ms across all node sizes. This is in stark contrast to PandaPower, which requires approximately 28 to 37 ms. In the smallest node size (25 nodes), Laurent is about 47 times faster than PandaPower when solving one-time power flow. As the node size grows to 123, the efficiency margin increases, with

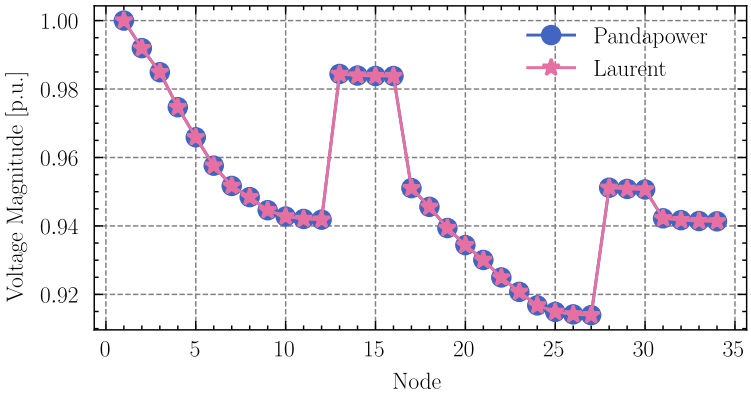


Figure 6.9: Voltage magnitude calculated by Laurent power flow and Panda power. The 34-node distribution network is used as a benchmark.

Table 6.4: Average calculation time comparison between Laurent power flow and PandaPower power flow for different scale distribution networks

Distribution Networks	**Laurent Power**		**PandaPower**	
	Power Flow $[ms]$	Env. Steps $[ms]$	Power Flow $[ms]$	Env. Steps $[ms]$
25 Nodes	**0.59**	2.81	28.08	30.30
34 Nodes	**0.61**	2.830	29.42	30.502
69 Nodes	**0.88**	2.99	28.72	31.46
123 Nodes	**0.97**	3.43	37.22	38.51

Laurent being nearly 38 times faster.

For executing one-time environment iteration, Laurent's time ranges from 2.8 to 3.4 ms, while PandaPower's duration extends from 30 to 38 ms. This indicates that, on average, Laurent is about ten times faster than PandaPower in processing environment steps, regardless of the node size. Overall, the Laurent power flow displays a significant computational edge, particularly as the node size expands. This relative efficiency is pivotal in training DRL algorithms in large-scale distribution networks. The ability of the Laurent power flow to consistently outpace PandaPower across different node sizes underscores its scalability, making it a more versatile choice for varied applications.

The comparison between Laurent power flow and PandaPower flow algorithms across different DRL algorithms showcases significant time differences in training for the same number of episodes as shown in Fig. 6.8. A clear trend emerges from the data: the Laurent power flow consistently outperforms PandaPower in terms of computational efficiency. For the SAC algorithm, the Laurent power flow is approximately 4.4 times faster than the PandaPower flow. Similarly, for DDPG, the Laurent method shows a speedup of around 5.2 times. The TD3 algorithm with the Laurent technology is about 4.8 times faster. The most pronounced difference is observed in the PPO algorithm, where Laurent power flow is significantly faster, clocking at approximately 9.1 times the speed of PandaPower. PPO requires 2200 minutes for training, making it the least efficient in this

scenario. This is because PPO is an off-policy algorithm, which can not fully make use of the past experiences in the replay buffer, resulting in the lowest data efficiency and training speed. On the other hand, DDPG emerges as the fastest, closely followed by TD3 and then SAC.

In conclusion, the Laurent power flow demonstrates a clear computational advantage across all tested algorithms. While the choice of algorithm also affects the training time, with PPO consistently taking the longest, the underlying power flow technology plays a crucial role in determining the overall efficiency. These findings can guide researchers and practitioners in making informed decisions when it comes to selecting the most efficient combination of power flow technology and reinforcement learning algorithm.

Fig. 6.9 displays the voltage magnitude results of a 34-node distribution network from Laurent power flow and PandaPower flow, respectively. The voltage magnitude results from both algorithms remain almost the same magnitude, with an average error of no more than 0.0001%. Such a high precision from Laurent power flow can track the real voltage dynamics accurately, regrading of the load changes. Moreover, the integration of Laurent power flow with the developed environment can significantly save the time cost for a large magnitude power flow iteration during the training. Thus, our framework can accelerate notably training speed of DRL algorithms, without losing simulation precision.

6.6. DISCUSSION

The RL-ADN environment offers enhanced flexibility and customization, surpassing existing frameworks like CityLearn and GYM-ANM, which exhibit limited adaptability in modeling complex distribution networks. CityLearn focuses on building-level energy management, simplifying grid-level dynamics, while GYM-ANM lacks precision for complex network modeling. These limitations restrict the effectiveness of RL agents in real-world deployment. In contrast, RL-ADN provides extensive customization options, allowing researchers to model complex network topologies, integrate diverse ESSs, and design tailored MDPs. This flexibility helps bridge the sim-to-real gap, as demonstrated by RL-ADN's ability to adapt to complex pricing and load conditions more effectively than traditional frameworks.

The proposed RL-ADN environment includes a data augmentation module based on a GMC approach, significantly enhancing training scenario diversity and improving DRL performance. Unlike other frameworks that converge to local optima due to limited data, RL-ADN enables agents to learn from a broader range of scenarios, resulting in more effective policies. This addresses a key limitation of frameworks like PowerGridWorld and Grid2OP, where limited data diversity restricts real-world applicability.

Existing environments, such as those using PandaPower, face high computational demands, reducing efficiency for DRL training. PandaPower-based solutions can take tens of milliseconds for each power flow iteration, becoming a bottleneck during training. RL-ADN integrates the Tensor Power Flow solver, which achieves a tenfold increase in speed compared to PandaPower, greatly accelerating DRL training without sacrificing accuracy. Fig. 6.9 shows that Tensor Power Flow results closely match those results from PandaPower, ensuring realistic and efficient training.

While RL-ADN demonstrates significant advancements, there are limitations to its current implementation. One key challenge is the gap between simulation and reality, as building an accurate distribution network simulator is difficult [188]. This can lead to discrepancies when deploying RL agents trained in simulation to real-world environments. Another limitation is the potential difficulty in extending RL-ADN to integrated energy systems, such as transportation or hydrogen networks [189]. These systems introduce additional layers of complexity and require further development to handle their unique dynamics and computational requirements. Future work will focus on addressing these limitations by enhancing the accuracy of the distribution network simulations and extending the framework to integrated energy systems, including transportation and hydrogen, to improve the applicability and robustness of RL-ADN in diverse real-world conditions.

Overall, RL-ADN sets a new benchmark in applying DRL to dispatch ESSs tasks in distribution networks, offering a comprehensive solution that addresses the limitations of existing environments.

7

Conclusion and Discussion

7.1. RESEARCH CONTRIBUTIONS TO RESEARCH QUESTIONS

The research questions proposed in Chapter 1 have been addressed through a combination of theoretical analysis and experimental validation. The main contributions and conclusions related to each research question are summarized as follows:

7.1.1. ENFORCING DISTRIBUTION NETWORK OPERATIONAL CONSTRAINTS USING DRL (Q1):

Chapter 2 introduced the MIP-DQN algorithm, a value-based Deep Reinforcement Learning (DRL) approach that integrates Mixed-Integer Linear Programming (MIP) to enforce power balance constraints in the operation of distributed energy resources (DERs). The proposed MIP-DQN algorithm ensures that all operational constraints, particularly power balance, are strictly enforced during online execution, thereby ensuring the feasibility of the dispatch schedule. The essence of the MIP-DQN algorithm lies in using a trained Q-network as a surrogate function for optimal operational decisions. This network is reformulated as a MIP to strictly enforce power balance constraints during real-time operation. The results demonstrated that MIP-DQN could achieve near-optimal solutions with a minimal error margin of 13.7% when compared with the optimal solution obtained using perfect forecasts. However, it was observed that the quality of the Q-network approximation directly impacts the performance of the algorithm, indicating a need for further improvements in training strategies and hyperparameter tuning. The MIP-DQN algorithm offers a robust solution to enforcing power balance constraints in real-time operations by combining deep learning with optimization techniques. Future work should focus on enhancing the Q-network's approximation quality and exploring alternative exploration strategies to improve the algorithm's overall performance.

Chapter 3 further developed the MIP-DRL framework, which extends the capabilities of standard actor-critic DRL algorithms by transforming the Q-network into a MIP formulation. This framework ensures strict enforcement of operational constraints, particularly in the dispatch of energy storage systems (ESSs) in distribution networks. A key insight from this work is the impact of the exploration-exploitation dilemma on the performance of different DRL algorithms within the MIP-DRL framework. For instance, the MIP-DDPG algorithm outperformed the MIP-TD3 algorithm, while the MIP-SAC algorithm performed conservatively due to its soft Q-updating rule. These findings suggest that the choice of exploration policy and Q-network update rules are critical factors influencing the effectiveness of the MIP-DRL framework. The MIP-DRL framework significantly enhances the enforcement of operational constraints in DRL algorithms by leveraging MIP formulations. However, the performance is closely tied to the quality of the Q-network and the chosen exploration strategy. Future work should focus on optimizing these aspects to further improve the applicability of the framework in complex energy systems.

7.1.2. LEVERAGING DOMAIN KNOWLEDGE FOR SAFETY, PERFORMANCE, AND COMPUTATIONAL EFFICIENCY (Q2):

Chapter 4 focused on enforcing voltage magnitude constraints within distribution networks using the DistFlow Safe Reinforcement Learning (DF-SRL) algorithm. The DF-SRL

algorithm incorporates expert knowledge into a safety layer that recalibrates potentially unsafe actions during both training and operational phases. This approach ensures that voltage magnitude constraints are strictly enforced, even under severe conditions such as extreme loading scenarios.

The sensitivity analysis conducted on the slack parameter ϵ revealed critical role of DF-SRL in balancing optimality and feasibility. An optimal value of $\epsilon = 0.002$ was found to provide the best trade-off, ensuring that voltage constraints were met without compromising performance. Additionally, the scalability of the DF-SRL algorithm was validated across various network sizes, demonstrating its robustness and applicability to large-scale distribution networks.

Chapter 5 further introduced a Safe Imitation Reinforcement Learning Framework that combines Twin Delayed Deep Deterministic Policy Gradient (TD3) with Inverse Reinforcement Learning (IRL). This framework addresses the challenge of ensuring safety and efficiency in DRL by incorporating a safe layer that filters unsafe actions during the learning process.

The framework demonstrated significant improvements in training efficiency and operational performance by leveraging expert data to guide the policy during training. However, it was observed that TD3BC, while effective in improving performance, struggled with ensuring safety in scenarios not covered by the training data, leading to potential operational hazards. The integration of the safe layer addressed this issue by ensuring that all actions complied with operational constraints, even in unseen scenarios.

The Safe Imitation Learning framework effectively enhances DRL algorithms by balancing performance improvements with rigorous safety enforcement. However, ensuring robustness against real-world scenarios that differ from the training data remains a challenge. Future research could focus on developing more adaptive safe layers and exploring hybrid learning approaches to further improve the framework's safety and efficiency.

7.1.3. REDUCING COMPUTATIONAL COST AND ACCELERATING TRAINING IN DRL ALGORITHMS (Q3):

Chapter 6 introduced RL-ADN, an open-source library designed to reduce computational costs and accelerate the training of DRL algorithms for optimal ESSs dispatch in distribution networks. RL-ADN integrates the Laurent power flow solver and Gaussian mixture models for data augmentation, significantly improving the training efficiency and performance of DRL algorithms.

A key innovation in RL-ADN is its ability to generate diverse training scenarios using advanced data augmentation techniques, thereby enhancing the performance ceiling of DRL algorithms. Additionally, the integration of the Laurent power flow solver provides a substantial reduction in computation time, making RL-ADN highly suitable for large-scale energy system applications.

Conclusion: RL-ADN represents a significant advancement in the development and deployment of DRL algorithms for energy systems. Its modular design, coupled with powerful computational techniques, offers a flexible and efficient platform for training DRL models. Future work could explore further enhancements to the data augmentation module and the integration of more sophisticated power flow solvers to continue

improving the capabilities of RL-ADN.

7.2. DISCUSSION AND RESEARCH RECOMMENDATIONS

The findings of this thesis suggest several avenues for future research to further enhance the applicability and performance of DRL algorithms in energy system operations. The detailed recommendations are as follows:

- **IMPROVING SCALABILITY AND EFFICIENCY:**
The scalability of MIP-based DRL frameworks remains a challenge due to the computational complexity of solving MIP formulations in large-scale systems. Future research should focus on developing more efficient optimization algorithms or approximation methods to reduce computation time while maintaining constraint enforcement.

- **HYBRID APPROACHES FOR ENHANCED EXPLORATION:**
Combining model-based predictions with model-free DRL could lead to more efficient exploration strategies, potentially improving convergence rates and overall performance in complex environments. Research into hybrid DRL frameworks could yield significant advancements in this area.

- **REAL-WORLD DEPLOYMENT AND ROBUSTNESS:**
The transition from simulation to real-world deployment of DRL algorithms presents numerous challenges, particularly in ensuring robustness against unforeseen events and variations in real-time data. Future work should focus on developing adaptive learning mechanisms that allow DRL algorithms to continuously update and refine their models based on real-world feedback.

- **ADVANCED SAFE LAYERS FOR SEQUENTIAL DECISION-MAKING:**
While current safe layer-based DRL algorithms effectively manage state-wise constraints, they often struggle with sequential decision-making processes. Research into developing more advanced safe layers that account for the cumulative and interdependent effects of sequential decisions could significantly improve the performance of DRL in complex operational settings.

- **MULTI-AGENT DRL SYSTEMS:**
The application of DRL in multi-agent systems, where multiple DERs and ESSs operate cooperatively, is an emerging area of interest. Future research could explore coordination mechanisms that ensure global optimality and stability across the entire energy system, leveraging the collective intelligence of multi-agent DRL frameworks.

BIBLIOGRAPHY

[1] Christian Breyer et al. “On the history and future of 100% renewable energy systems research”. In: *IEEE Access* 10 (2022), pp. 78176–78218.

[2] Dolf Gielen et al. “The role of renewable energy in the global energy transformation”. In: *Energy strategy reviews* 24 (2019), pp. 38–50.

[3] John Twidell. *Renewable energy resources*. Routledge, 2021.

[4] Wei Sun, Amir Golshani, et al. “Distributed restoration for integrated transmission and distribution systems with DERs”. In: *IEEE Transactions on Power Systems* 34.6 (2019), pp. 4964–4973.

[5] Ateeq Ur Rehman et al. “An optimal power usage scheduling in smart grid integrated with renewable energy sources for energy management”. In: *IEEE Access* 9 (2021), pp. 84619–84638.

[6] Yongli Wang et al. “Optimal scheduling of the regional integrated energy system considering economy and environment”. In: *IEEE Transactions on Sustainable Energy* 10.4 (2018), pp. 1939–1949.

[7] Yuqing Yang et al. “Modelling and optimal energy management for battery energy storage systems in renewable energy systems: A review”. In: *Renewable and Sustainable Energy Reviews* 167 (2022), p. 112671.

[8] Carlos Medina, C Ríos M Ana, and Guadalupe González. “Transmission grids to foster high penetration of large-scale variable renewable energy sources–A review of challenges, problems, and solutions”. In: *International Journal of Renewable Energy Research (IJRER)* 12.1 (2022), pp. 146–169.

[9] Sumit K Rathor and Dipti Saxena. “Energy management system for smart grid: An overview and key issues”. In: *International Journal of Energy Research* 44.6 (2020), pp. 4067–4109.

[10] Jingda Wu et al. “Confidence-aware reinforcement learning for energy management of electrified vehicles”. In: *Renewable and Sustainable Energy Reviews* 191 (2024), p. 114154.

[11] Haifeng Qiu et al. “Application of two-stage robust optimization theory in power system scheduling under uncertainties: A review and perspective”. In: *Energy* 251 (2022), p. 123942.

[12] Xin Chen et al. “Reinforcement learning for decision-making and control in power systems: Tutorial, review, and vision”. In: *arXiv preprint arXiv:2102.01168* (2021).

[13] Richard S Sutton and Andrew G Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

[14] Mengfan Zhang et al. “Data driven decentralized control of inverter based renewable energy sources using safe guaranteed multi-agent deep reinforcement learning”. In: *IEEE Transactions on Sustainable Energy* (2023).

[15] N Bhavatarini, Syed Thouheed Ahmed, and Syed Muzamil Basha. *Reinforcement Learning-Principles, Concepts and Applications*. MileStone Research Publications, 2024.

[16] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, et al. “Human-level control through deep reinforcement learning”. In: *Nature* 518.7540 (2015), pp. 529–533.

[17] Hado Van Hasselt, Arthur Guez, and David Silver. “Deep reinforcement learning with double q-learning”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 30. 1. 2016.

[18] Ziyu Wang et al. “Dueling network architectures for deep reinforcement learning”. In: *International conference on machine learning*. PMLR. 2016, pp. 1995–2003.

[19] Matteo Hessel et al. “Rainbow: Combining improvements in deep reinforcement learning”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 1. 2018.

[20] Tom Schaul. “Prioritized Experience Replay”. In: *arXiv preprint arXiv:1511.05952* (2015).

[21] Richard S Sutton, Satinder Singh, and David McAllester. “Comparing policy-gradient algorithms”. In: *IEEE Transactions on Systems, Man, and Cybernetics* (2000).

[22] John Schulman, Filip Wolski, Prafulla Dhariwal, et al. “Proximal policy optimization algorithms”. In: *arXiv preprint arXiv:1707.06347* (2017).

[23] T Lilicrap, J Hunt, Alexander Pritzel, et al. “Continuous control with deep reinforcement learning”. In: *International Conference on Representation Learning (ICRL)*. 2016.

[24] Scott Fujimoto, Herke Hoof, and David Meger. “Addressing function approximation error in actor-critic methods”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 1587–1596.

[25] Ying Ji et al. “Real-time energy management of a microgrid using deep reinforcement learning”. In: *Energies* 12.12 (2019), p. 2291.

[26] Jianhong Wang, Wangkun Xu, Yunjie Gu, et al. “Multi-agent reinforcement learning for active voltage control on power distribution networks”. In: *Advances in Neural Information Processing Systems* 34 (2021), pp. 3271–3284.

[27] Ting Yang et al. “Reinforcement learning in sustainable energy and electric systems: A survey”. In: *Annual Reviews in Control* 49 (2020), pp. 145–163.

[28] Peipei Yu et al. “Safe Reinforcement Learning for Power System Control: A Review”. In: *arXiv preprint arXiv:2407.00681* (2024).

[29] Pierre-Franx00E7;ois Massiani et al. “Safe Value Functions”. In: *IEEE Transactions on Automatic Control* (2022), pp. 1–16.

[30] Hou Shengren, Edgar Mauricio Salazar, Pedro P Vergara, et al. “Performance comparison of deep RL algorithms for energy systems optimal scheduling”. In: *2022 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*. IEEE. 2022, pp. 1–6.

[31] Edgar Mauricio Salazar Duque et al. “Tensor Power Flow Formulations for Multidimensional Analyses in Distribution Systems”. In: *arXiv preprint arXiv:2403.04578* (2024).

[32] Shengren Hou, Edgar Mauricio Salazar Duque, Peter Palensky, et al. “A Constraint Enforcement Deep Reinforcement Learning Framework for Optimal Energy Storage Systems Dispatch”. In: *arXiv preprint arXiv:2307.14304* (2023).

[33] Bei Li and Robin Roche. “Optimal scheduling of multiple multi-energy supply microgrids considering future prediction impacts based on model predictive control”. In: *Energy* 197 (2020), p. 117180.

[34] Antonio Carlos Zambroni de Souza and Miguel Castilla. *Microgrids design and implementation*. Springer, 2019.

[35] Pedro P Vergara et al. “Optimal dispatch of PV inverters in unbalanced distribution systems using Reinforcement Learning”. In: *International Journal of Electrical Power & Energy Systems* 136 (2022), p. 107628.

[36] Juan S. Giraldo et al. “Microgrids Energy Management Using Robust Convex Programming”. In: *IEEE Transactions on Smart Grid* 10.4 (2019), pp. 4520–4530.

[37] Mojtaba Yousefi, Amin Hajizadeh, and Mohsen Nourbakhsh Soltani. “A Comparison Study on Stochastic Modeling Methods for Home Energy Management Systems”. In: *IEEE Trans. Industrial Informatics* 15.8 (2019), pp. 4799–4808.

[38] Mojtaba Yousefi et al. “Predictive Home Energy Management System With Photovoltaic Array, Heat Pump, and Plug-In Electric Vehicle”. In: *IEEE Trans. Industrial Informatics* 17.1 (2021), pp. 430–440.

[39] Pedro P Vergara et al. “A stochastic programming model for the optimal operation of unbalanced three-phase islanded microgrids”. In: *International Journal of Electrical Power & Energy Systems* 115 (2020), p. 105446.

[40] Javier Arroyo et al. “Reinforced model predictive control (RL-MPC) for building energy management”. In: *Applied Energy* 309 (2022), p. 118346.

[41] Lingmin Chen et al. “A robust optimization framework for energy management of CCHP users with integrated demand response in electricity market”. In: *International Journal of Electrical Power & Energy Systems* 141 (2022), p. 108181.

[42] Su Su et al. “Energy management for active distribution network incorporating office buildings based on chance-constrained programming”. In: *International Journal of Electrical Power & Energy Systems* 134 (2022), p. 107360.

[43] Glenn Ceusters et al. “Model-predictive control and reinforcement learning in multi-energy system case studies”. In: *Applied Energy* 303 (2021), p. 117634.

[44] Hao Dong et al. *Deep Reinforcement Learning*. Springer, 2020.

[45] ATD Perera and Parameswaran Kamalaruban. “Applications of reinforcement learning in energy systems”. In: *Renewable and Sustainable Energy Reviews* 137 (2021), p. 110618.

[46] Taha Abdelhalim Nakabi and Pekka Toivanen. “Deep reinforcement learning for energy management in a microgrid with flexible demand”. In: *Sustainable Energy, Grids and Networks* 25 (2021), p. 100413.

[47] Qinglin Meng et al. “An online reinforcement learning-based energy management strategy for microgrids with centralized control”. In: *IEEE Transactions on Industry Applications* (2024).

[48] Adrian Kelly et al. “Reinforcement learning for electricity network operation”. In: *arXiv preprint arXiv:2003.07339* (2020).

[49] Liang Yu et al. “A review of deep reinforcement learning for smart building energy management”. In: *IEEE Internet of Things Journal* 8.15 (2021), pp. 12046–12063.

[50] Yuhao Zhou et al. “A data-driven method for fast ac optimal power flow solutions via deep reinforcement learning”. In: *Journal of Modern Power Systems and Clean Energy* 8.6 (2020), pp. 1128–1139.

[51] Giuseppe Pinto, Davide Deltetto, and Alfonso Capozzoli. “Data-driven district energy management with surrogate models and deep reinforcement learning”. In: *Applied Energy* 304 (2021), p. 117642.

[52] José R Vázquez-Canteli et al. “CityLearn: Standardizing research in multi-agent reinforcement learning for demand response and urban energy management”. In: *arXiv preprint arXiv:2012.10504* (2020).

[53] Amirreza Heidari, François Maréchal, and Dolaana Khovalyg. “Reinforcement Learning for proactive operation of residential energy systems by learning stochastic occupant behavior and fluctuating solar energy: Balancing comfort, hygiene and energy use”. In: *Applied Energy* 318 (2022), p. 119206.

[54] Linpeng Liu et al. “Deep Reinforcement Learning for Stochastic Dynamic Microgrid Energy Management”. In: *2021 IEEE 4th International Electrical and Energy Conference (CIEEC)*. IEEE. 2021, pp. 1–6.

[55] Adam Stooke, Joshua Achiam, and Pieter Abbeel. “Responsive safety in reinforcement learning by pid lagrangian methods”. In: *International Conference on Machine Learning*. PMLR. 2020, pp. 9133–9143.

[56] Suyang Zhou et al. “Combined heat and power system intelligent economic dispatch: A deep reinforcement learning approach”. In: *International journal of electrical power & energy systems* 120 (2020), p. 106016.

[57] Ying Ji et al. “Data-Driven Online Energy Scheduling of a Microgrid Based on Deep Reinforcement Learning”. In: *Energies* 14.8 (2021), p. 2120.

[58] Pedro P Vergara, Mauricio Salazar, Juan S Giraldo, et al. “Optimal dispatch of PV inverters in unbalanced distribution systems using Reinforcement Learning”. In: *Int. J. of Elec. Power & Energy Systems* 136 (2022), p. 107628.

[59] Edgar Mauricio Salazar Duque et al. “Community energy storage operation via reinforcement learning with eligibility traces”. In: *Electric Power Systems Research* 212 (2022), p. 108515. ISSN: 0378-7796.

[60] Weirong Liu et al. “Distributed economic dispatch in microgrids based on cooperative reinforcement learning”. In: *IEEE Trans. Neural Networks and Learning Systems* 29.6 (2018), pp. 2192–2203.

[61] Yan Du and Di Wu. “Deep Reinforcement Learning From Demonstrations to Assist Service Restoration in Islanded Microgrids”. In: *IEEE Transactions on Sustainable Energy* 13.2 (2022), pp. 1062–1072.

[62] Dawei Qiu et al. “Coordination for Multi-Energy Microgrids Using Multi-Agent Reinforcement Learning”. In: *IEEE Transactions on Industrial Informatics* (2022).

[63] Zhongkai Yi et al. “An Improved Two-Stage Deep Reinforcement Learning Approach for Regulation Service Disaggregation in a Virtual Power Plant”. In: *IEEE Transactions on Smart Grid* (2022).

[64] Bo Hu and Jiaxi Li. “Shifting Deep Reinforcement Learning Algorithm Toward Training Directly in Transient Real-World Environment: A Case Study in Powertrain Control”. In: *IEEE Trans. Industrial Informatics* 17.12 (2021), pp. 8198–8206.

[65] Javier Garcıa and Fernando Fernández. “A comprehensive survey on safe reinforcement learning”. In: *Journal of Machine Learning Research* 16.1 (2015), pp. 1437–1480.

[66] Mohammed Alshiekh et al. “Safe reinforcement learning via shielding”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 32. 1. 2018.

[67] Nils Jansen et al. “Shielded decision-making in MDPs”. In: *arXiv preprint arXiv:1807.06096* (2018).

[68] Somil Bharadwaj et al. “Conservative safety critics for exploration”. In: *arXiv preprint arXiv:2010.14497* (2020).

[69] Qisong Yang et al. “WCSAC: Worst-Case Soft Actor Critic for Safety-Constrained Reinforcement Learning”. In: *Proceedings of the AAAI Conference on Artificial Intelligence*. Vol. 35. 12. 2021, pp. 10639–10646.

[70] Michael Eichelbeck, Hannah Markgraf, and Matthias Althoff. “Contingency constrained economic dispatch with safe reinforcement learning”. In: *2022 21st IEEE International Conference on Machine Learning and Applications (ICMLA)*. IEEE. 2022, pp. 597–602.

[71] Sebastien Gros, Mario Zanon, and Alberto Bemporad. “Safe reinforcement learning via projection on a safe set: How to achieve optimality?” In: *IFAC-PapersOnLine* 53.2 (2020), pp. 8076–8081.

[72] Dawei Qiu et al. “Safe reinforcement learning for real-time automatic control in a smart energy-hub”. In: *Applied Energy* 309 (2022), p. 118403.

[73] Hyungjun Park et al. “DIP-QL: A Novel Reinforcement Learning Method for Constrained Industrial Systems”. In: *IEEE Trans. on Industrial Informatics* (2022).

[74] Hepeng Li and Haibo He. “Learning to Operate Distribution Networks With Safe Deep Reinforcement Learning”. In: *IEEE Trans. Smart Grid* 13.3 (2022), pp. 1860–1872.

[75] Hepeng Li, Zhiqiang Wan, and Haibo He. “Constrained EV charging scheduling based on safe deep reinforcement learning”. In: *IEEE Trans. on Smart Grid* 11.3 (2019), pp. 2427–2439.

[76] Matteo Fischetti and Jason Jo. “Deep neural networks and mixed integer linear optimization”. In: *Constraints*. Vol. 23. 2018, pp. 296–309.

[77] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, et al. “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.

[78] Awol Seid Ebrie and Young Jin Kim. “Reinforcement learning-based optimization for power scheduling in a renewable energy connected grid”. In: *Renewable Energy* 230 (2024), p. 120886.

[79] Christopher JCH Watkins and Peter Dayan. “Q-learning”. In: *Machine learning* 8.3 (1992), pp. 279–292.

[80] Moonkyung Ryu, Yinlam Chow, Ross Anderson, et al. “CAQL: Continuous Action Q-Learning”. In: *International Conference on Learning Representations*. 2020.

[81] Francesco Ceccon, Jordan Jalving, Joshua Haddad, et al. “OMLT: Optimization & Machine Learning Toolkit”. In: *arXiv preprint arXiv:2202.02414* (2022).

[82] Hou Shengren. https://github.com/ShengrenHou/DF-SRL. 2023.

[83] Chenyu Guo et al. “Optimal energy management of multi-microgrids connected to distribution system based on deep reinforcement learning”. In: *International journal of electrical power & energy systems* 131 (2021), p. 107048.

[84] William E Hart, Carl D Laird, Jean-Paul Watson, et al. *Pyomo-optimization modeling in python*. Vol. 67. Springer, 2017.

[85] Gal Dalal, Krishnamurthy Dvijotham, Matej Vecerik, et al. “Safe exploration in continuous action spaces”. In: *arXiv preprint arXiv:1801.08757* (Jan. 2018).

[86] Alex Ray, Joshua Achiam, and Dario Amodei. “Benchmarking safe exploration in deep reinforcement learning”. In: *arXiv preprint arXiv:1910.01708* 7 (2019), p. 1.

[87] Shengren Hou et al. “A Mix-Integer Programming Based Deep Reinforcement Learning Framework for Optimal Dispatch of Energy Storage System in Distribution Networks”. In: *Journal of Modern Power Systems and Clean Energy* (2024).

[88] Osama Majeed Butt, Muhammad Zulqarnain, and Tallal Majeed Butt. “Recent advancement in smart grid technology: Future prospects in the electrical power network”. In: *Ain Shams Engineering Journal* 12.1 (2021), pp. 687–695.

[89] Víctor M Garrido-Arévalo et al. “Optimal Dispatch of DERs and Battery-Based ESS in Distribution Grids While Considering Reactive Power Capabilities and Uncertainties: A Second-Order Cone Programming Formulation”. In: *IEEE Access* (2024).

[90] Huilong Yu et al. “Mixed-integer optimal design and energy management of hybrid electric vehicles with automated manual transmissions”. In: *IEEE Transactions on Vehicular Technology* 69.11 (2020), pp. 12705–12715.

[91] Farid Hamzeh Aghdam et al. “Optimal scheduling of multi-energy type virtual energy storage system in reconfigurable distribution networks for congestion management”. In: *Applied Energy* 333 (2023), p. 120569. ISSN: 0306-2619.

[92] Di Cao, Weihao Hu, Jun.bo Zhao, et al. “Reinforcement learning and its applications in modern power and energy systems: A review”. In: *Journal of Modern Power Systems and Clean Energy* 8.6 (Nov. 2020), pp. 1029–1042.

[93] Ziyang Yin, Shouxiang Wang, and Qianyu Zhao. “Sequential Reconfiguration of Unbalanced Distribution Network with Soft Open Points Based on Deep Reinforcement Learning”. In: *Journal of Modern Power Systems and Clean Energy* 11.1 (Jan. 2022), pp. 107–119.

[94] Igor Halperin. *Reinforcement Learning and Stochastic Optimization: A Unified Framework for Sequential Decisions: by Warren B. Powell (ed.), Wiley (2022). Hardback. ISBN 9781119815051*. Vol. 22. 12. Taylor & Francis, 2022.

[95] Jonas Degrave, Federico Felici, Jonas Buchli, et al. “Magnetic control of tokamak plasmas through deep reinforcement learning”. In: *Nature* 602.7897 (Feb. 2022), pp. 414–419.

[96] Yan Du et al. “Demonstration of Intelligent HVAC Load Management With Deep Reinforcement Learning: Real-World Experience of Machine Learning in Demand Control”. In: *IEEE Power and Energy Magazine* 20.3 (2022), pp. 42–53.

[97] Haeun Yoo, Victor M Zavala, and Jay H Lee. “A dynamic penalty function approach for constraint-handling in reinforcement learning”. In: *IFAC-PapersOnLine* 54.3 (2021), pp. 487–491.

[98] Xiaofeng Yang et al. “Enabling Safety-Enhanced fast charging of electric vehicles via soft actor Critic-Lagrange DRL algorithm in a Cyber-Physical system”. In: *Applied Energy* 329 (Feb. 2023), p. 120272.

[99] Han Cui, Yujian Ye, Jianxiong Hu, et al. “Online Preventive Control for Transmission Overload Relief Using Safe Reinforcement Learning with Enhanced Spatial-Temporal Awareness”. In: *IEEE Transactions on Power Systems* (Feb. 2023).

[100] Joshua Achiam, David Held, Aviv Tamar, et al. “Constrained policy optimization”. In: *International conference on machine learning*. PMLR. 2017, pp. 22–31.

[101] Glenn Ceusters et al. “An adaptive safety layer with hard constraints for safe reinforcement learning in multi-energy management systems”. In: *Sustainable Energy, Grids and Networks* 36 (Feb. 2023), p. 101202.

[102] Sebastien Gros, Mario Zanon, and Alberto Bemporad. “Safe reinforcement learning via projection on a safe set: How to achieve optimality?” In: *IFAC-PapersOnLine* 53.2 (2020), pp. 8076–8081.

[103] Peng Kou, Deliang Liang, Chen Wang, et al. “Safe deep reinforcement learning-based constrained optimal control scheme for active distribution networks”. In: *Applied energy* 264 (2020), p. 114772.

[104] Shangding Gu et al. “A review of safe reinforcement learning: Methods, theory and applications”. In: *arXiv preprint arXiv:2205.10330* (2022).

[105] Hou Shengren, Pedro P. Vergara, Edgar Mauricio Salazar Duque, et al. “Optimal Energy System Scheduling Using a Constraint-Aware Reinforcement Learning Algorithm”. In: *International Journal of Electrical Power & Energy Systems* 152 (Oct. 2023), p. 109230. ISSN: 0142-0615. (Visited on 06/08/2023).

[106] Yang Xia, Yan Xu, and Xue Feng. “Hierarchical Coordination of Networked Microgrids towards Decentralized Operation: A Safe Deep Reinforcement Learning Method”. In: *IEEE Transactions on Sustainable Energy* (2024).

[107] Hongyuan Ding et al. “A safe reinforcement learning approach for multi-energy management of smart home”. In: *Electric Power Systems Research* 210 (Nov. 2022), p. 108120.

[108] Shulei Zhang et al. “A safe reinforcement learning-based charging strategy for electric vehicles in residential microgrid”. In: *Applied Energy* 348 (May 2023), p. 121490.

[109] Yujian Ye et al. “Safe Deep Reinforcement Learning for Microgrid Energy Management in Distribution Networks with Leveraged Spatial-Temporal Perception”. In: *IEEE Transactions on Smart Grid* (July 2023).

[110] Peipei Yu, Hongcai Zhang, and Yonghua Song. “District cooling system control for providing regulation services based on safe reinforcement learning with barrier functions”. In: *Applied Energy* 347 (Sept. 2023), p. 121396.

[111] Mohammad Mehdi Hosseini and Masood Parvania. “On the Feasibility Guarantees of Deep Reinforcement Learning Solutions for Distribution System Operation”. In: *IEEE Transactions on Smart Grid* 14.2 (Mar. 2023), pp. 954–964.

[112] Yuanyuan Shi et al. “Stability constrained reinforcement learning for real-time voltage control”. In: *2022 American Control Conference (ACC)*. IEEE. Atlanta, GA, USA, June 2022, pp. 2715–2721.

[113] Leonardo H. Macedo et al. “Optimal Operation of Distribution Networks Considering Energy Storage Devices”. In: *IEEE Transactions on Smart Grid* 6.6 (Nov. 2015), pp. 2825–2836.

[114] Sungsu Lim et al. “Actor-expert: A framework for using q-learning in continuous action spaces”. In: *arXiv preprint arXiv:1810.09103* (2018).

[115] Guido F Montufar et al. “On the number of linear regions of deep neural networks”. In: *Advances in Neural Information Processing Systems* 27 (Dec. 2014).

[116] *What’s New – Gurobi 10.0*. en-US. URL: https://www.gurobi.com/whats-new-gurobi-10-0/ (visited on 06/07/2023).

[117] Tianhao Wei and Changliu Liu. “Safe control with neural network dynamic models”. In: *Learning for Dynamics and Control Conference*. PMLR. Berkeley, CA, USA, July 2022, pp. 739–750.

[118] Shengren Hou et al. “DistFlow Safe Reinforcement Learning Algorithm for Voltage Magnitude Regulation in Distribution Networks”. In: *Journal of Modern Power Systems and Clean Energy* (2024).

[119] Tayenne Dias de Lima et al. “Modern distribution system expansion planning considering new market designs: Review and future directions”. In: *Renewable and Sustainable Energy Reviews* 202 (2024), p. 114709.

[120] Rahul K Gupta, Paprapee Buason, and Daniel K Molzahn. “Fairness-aware photovoltaic generation limits for voltage regulation in power distribution networks using conservative linear approximations”. In: *2024 IEEE Texas Power and Energy Conference (TPEC)*. IEEE. 2024, pp. 1–6.

[121] Xin Chen, Emiliano Dall’Anese, Changhong Zhao, et al. “Aggregate power flexibility in unbalanced distribution systems”. In: *IEEE Transactions on Smart Grid* 11.1 (Jan. 2019), pp. 258–269.

[122] Chuyi Li et al. “Intra-day optimal power flow considering flexible workload scheduling of IDCs”. In: *Energy Reports* 9 (May 2023), pp. 1149–1159.

[123] Yinxiao Li et al. “Optimal Dispatch of Battery Energy Storage in Distribution Network Considering Electrothermal-Aging Coupling”. In: *IEEE Transactions on Smart Grid* (Jan. 2023).

[124] Mevludin Glavic. “(Deep) reinforcement learning for electric power system control and related problems: A short review and perspectives”. In: *Annual Reviews in Control* 48 (Apr. 2019), pp. 22–35.

[125] Glenn Ceusters et al. “Safe reinforcement learning for multi-energy management systems with known constraint functions”. In: *Energy and AI* 12 (2023), p. 100227.

[126] Chao Huang, Hongcai Zhang, Long Wang, et al. “Mixed deep reinforcement learning considering discrete-continuous hybrid action space for smart home energy management”. In: *Journal of Modern Power Systems and Clean Energy* 10.3 (May 2022), pp. 743–754.

[127] Shengyi Wang, JiaJun. Duan, Di Shi, et al. “A data-driven multi-agent autonomous voltage control framework using deep reinforcement learning”. In: *IEEE Transactions on Power Systems* 35.6 (Dec. 2020), pp. 4644–4654.

[128] Yongdong Chen et al. “Multiagent Soft Actor–Critic Learning for Distributed ESS Enabled Robust Voltage Regulation of Active Distribution Grids”. In: *IEEE Transactions on Industrial Informatics* (2024).

[129] Kai-Chieh Hsu et al. “Sim-to-lab-to-real: Safe reinforcement learning with shielding and generalization guarantees”. In: *Artificial Intelligence* 314 (2023), p. 103811.

[130] Weiye Zhao et al. “State-wise Safe Reinforcement Learning: A Survey”. In: *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*. IJCAI-2023. International Joint Conferences on Artificial Intelligence Organization, Aug. 2023.

[131] Wenqi Cui, Jiayi Li, and Baosen Zhang. “Decentralized safe reinforcement learning for inverter-based voltage control”. In: *Electric Power Systems Research* 211 (Dec. 2022), p. 108609.

[132] Wei Wang et al. “Safe Off-Policy Deep Reinforcement Learning Algorithm for Volt-VAR Control in Power Distribution Systems”. In: *IEEE Transactions on Smart Grid* 11.4 (Apr. 2020), pp. 3008–3018.

[133] Mengfan Zhang et al. “DNN Assisted Projection based Deep Reinforcement Learning for Safe Control of Distribution Grids”. In: *IEEE Transactions on Power Systems* (June 2023), pp. 1–12.

[134] Tu-Hoa Pham, Giovanni De Magistris, and Ryuki Tachibana. “OptLayer - Practical Constrained Optimization for Deep Reinforcement Learning in the Real World”. In: *2018 IEEE International Conference on Robotics and Automation (ICRA)*. Brisbane, Australia, May 2018, pp. 6236–6243.

[135] Edgar D Klenske and Philipp Hennig. “Dual control for approximate Bayesian reinforcement learning”. In: *Journal of Machine Learning Research* 17.127 (Aug. 2016), pp. 1–30.

[136] Xiao Shun Zhang et al. “Lifelong learning for complementary generation control of interconnected power grids with high-penetration renewables and EVs”. In: *IEEE Transactions on Power Systems* 33.4 (July 2017), pp. 4097–4110.

[137] Priyanka Chaudhary and M Rizwan. “Voltage regulation mitigation techniques in distribution system with high PV penetration: A review”. In: *Renewable and Sustainable Energy Reviews* 82 (2018), pp. 3279–3287.

[138] Xuewei Huang et al. “Research on Aggregation Flexibility Method of AC/DC Distribution Network Considering Flexibility Balance and Its Application”. In: *IEEE Systems Journal* 17.3 (2023), pp. 3635–3645.

[139] R Radhamani et al. “Deployment of an IoT-Integrated Home Energy Management System Employing Deep Reinforcement Learning”. In: *2024 2nd International Conference on Artificial Intelligence and Machine Learning Applications Theme: Healthcare and Internet of Things (AIMLA)*. IEEE. 2024, pp. 1–4.

[140] Eran Schweitzer, Shammya Saha, Anna Scaglione, et al. “Lossy distflow formulation for single and multiphase radial feeders”. In: *IEEE Transactions on Power Systems* 35.3 (May 2019), pp. 1758–1768.

[141] Pawel Ladosz et al. “Exploration in deep reinforcement learning: A survey”. In: *Information Fusion* 85 (2022), pp. 1–22.

[142] Aihui Fu, Miloš Cvetković, and Peter Palensky. “Distributed cooperation for voltage regulation in future distribution networks”. In: *IEEE Transactions on Smart Grid* 13.6 (Nov. 2022), pp. 4483–4493.

[143] Statistics Netherlands (CBS). *Power from solar panels increased slightly in 2023*. https://www.cbs.nl/en-gb/news/2024/25/power-from-solar-panels-increased-slightly-in-2023. [Accessed June, 2024]. 2024.

[144] Werner van Westering and Hans Hellendoorn. “Low voltage power grid congestion reduction using a community battery: Design principles, control and experimental validation”. In: *International Journal of Electrical Power & Energy Systems* 114 (2020), p. 105349.

[145] Vasko Zdraveski et al. “Radial distribution network planning under uncertainty by implementing robust optimization”. In: *International Journal of Electrical Power & Energy Systems* 149 (2023), p. 109043.

[146] Georgios C Kryonidis, Charis S Demoulias, and Grigoris K Papagiannis. “A two-stage solution to the bi-objective optimal voltage regulation problem”. In: *IEEE Transactions on Sustainable Energy* 11.2 (2019), pp. 928–937.

[147] Salish Maharjan, Ashwin M Khambadkone, and Jimmy Chih-Hsien Peng. “Robust constrained model predictive voltage control in active distribution networks”. In: *IEEE Transactions on Sustainable Energy* 12.1 (2020), pp. 400–411.

[148] Lucheng Hong et al. “MADRL-Based DSO-Customer Coordinated Bi-Level Volt/VAR Optimization Method for Power Distribution Networks”. In: *IEEE Transactions on Sustainable Energy* (2024).

[149] Rafael Ris-Ala. *Fundamentals of Reinforcement Learning*. Springer, 2023.

[150] Ramij Raja Hossain et al. “Efficient learning of power grid voltage control strategies via model-based deep reinforcement learning”. In: *Machine Learning* 113.5 (2024), pp. 2675–2700.

[151] Hongwen He et al. “Deep reinforcement learning based energy management strategies for electrified vehicles: Recent advances and perspectives”. In: *Renewable and Sustainable Energy Reviews* 192 (2024), p. 114248.

[152] Roshni Anna Jacob et al. “Real-time outage management in active distribution networks using reinforcement learning over graphs”. In: *Nature Communications* 15.1 (2024), p. 4766.

[153] Jichen Zhang et al. “Networked Multiagent-Based Safe Reinforcement Learning for Low-Carbon Demand Management in Distribution Networks”. In: *IEEE Transactions on Sustainable Energy* (2024).

[154] Xiao Zhang et al. “Application and progress of artificial intelligence technology in the field of distribution network voltage Control: A review”. In: *Renewable and Sustainable Energy Reviews* 192 (2024), p. 114282.

[155] Boyuan Zheng et al. “Imitation learning: Progress, taxonomies and challenges”. In: *IEEE Transactions on Neural Networks and Learning Systems* (2022).

[156] Vinicius G Goecks et al. “Integrating behavior cloning and reinforcement learning for improved performance in dense and sparse reward environments”. In: *arXiv preprint arXiv:1910.04281* (2019).

[157] Siyuan Guo et al. “Sample efficient offline-to-online reinforcement learning”. In: *IEEE Transactions on Knowledge and Data Engineering* (2023).

[158] Liangcheng Cheng et al. “Real-time dispatch via expert knowledge driven deep reinforcement learning”. In: *CSEE Journal of Power and Energy Systems* (2023).

[159] Huy Truong Dinh and Daehee Kim. “MILP-based imitation learning for HVAC control”. In: *IEEE Internet of Things Journal* 9.8 (2021), pp. 6107–6120.

[160] Shichao Xu et al. “Accelerate online reinforcement learning for building HVAC control with heterogeneous expert guidances”. In: *Proceedings of the 9th ACM International Conference on Systems for Energy-Efficient Buildings, Cities, and Transportation*. 2022, pp. 89–98.

[161] Shuhua Gao et al. “Online optimal power scheduling of a microgrid via imitation learning”. In: *IEEE Transactions on Smart Grid* 13.2 (2021), pp. 861–876.

[162] Scott Fujimoto and Shixiang Shane Gu. “A minimalist approach to offline reinforcement learning”. In: *Advances in neural information processing systems* 34 (2021), pp. 20132–20145.

[163] Shengren Hou et al. “A Mix-Integer Programming Based Deep Reinforcement Learning Framework for Optimal Dispatch of Energy Storage System in Distribution Networks”. In: *Journal of Modern Power Systems and Clean Energy* (2024).

[164] Shengren Hou et al. “RL-ADN: A High-Performance Deep Reinforcement Learning Environment for Optimal Energy Storage Systems Dispatch in Active Distribution Networks”. In: *arXiv preprint arXiv:2408.03685* (2024).

[165] Jan Martin Specht and Reinhard Madlener. “Deep reinforcement learning for the optimized operation of large amounts of distributed renewable energy assets”. In: *Energy and AI* 11 (2023), p. 100215.

[166] Pedro P. Vergara, Juan C. López, Marcos J. Rider, et al. “Optimal Operation of Unbalanced Three-Phase Islanded Droop-Based Microgrids”. In: *IEEE Trans. Smart Grid* 10.1 (2019), pp. 928–940.

[167] Hou Shengren. https://github.com/ShengrenHou/DF-SRL. 2023.

[168] Greg Brockman et al. “Openai gym”. In: *arXiv preprint arXiv:1606.01540* (2016).

[169] Elia Kaufmann et al. “Champion-level drone racing using deep reinforcement learning”. In: *Nature* 620.7976 (2023), pp. 982–987.

[170] Oguzhan Dogru et al. “Reinforcement Learning in Process Industries: Review and Perspective”. In: *IEEE/CAA Journal of Automatica Sinica* 11.2 (2024), pp. 283–300.

[171] Fernando Gallego et al. “Maintaining flexibility in smart grid consumption through deep learning and deep reinforcement learning”. In: *Energy and AI* 13 (2023), p. 100241.

[172] Stavros Karagiannopoulos et al. “Decentralized control in active distribution grids via supervised and reinforcement learning”. In: *Energy and AI* 16 (2024), p. 100342.

[173] Hantao Cui and Yichen Zhang. “Andes_gym: A versatile environment for deep reinforcement learning in power systems”. In: *2022 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE. 2022, pp. 01–05.

[174] Aisling Pigott et al. “GridLearn: Multiagent reinforcement learning for grid-aware building energy management”. In: *Electric Power Systems Research* 213 (2022), p. 108521.

[175] David Biagioni et al. “Powergridworld: A framework for multi-agent reinforcement learning in power systems”. In: *Proceedings of the Thirteenth ACM International Conference on Future Energy Systems*. 2022, pp. 565–570.

[176] B. Donnot. *Grid2op- A testbed platform to model sequential decision making in power systems*. https://GitHub.com/rte-france/grid2op. 2020.

[177] Robin Henry and Damien Ernst. “Gym-ANM: Reinforcement learning environments for active network management tasks in electricity distribution systems”. In: *Energy and AI* 5 (2021), p. 100092.

[178] Weijie Xia et al. “Comparative assessment of generative models for transformer- and consumer-level load profiles generation”. In: *Sustainable Energy, Grids and Networks* 38 (2024), p. 101338.

[179] Jingyi Xu et al. “A Comparative Study of Deep Reinforcement Learning-based Transferable Energy Management Strategies for Hybrid Electric Vehicles”. In: *2022 IEEE Intelligent Vehicles Symposium (IV)*. 2022, pp. 470–477.

[180] Henrik Bode et al. *Towards a Scalable and Flexible Simulation and Testing Environment Toolbox for Intelligent Microgrid Control*. 2020. arXiv: 2005.04869 [eess.SY].

[181] Marvin Lerousseau. “Design and implementation of an environment for Learning to Run a Power Network (L2RPN)”. In: *arXiv preprint arXiv:2104.04080* (2021).

[182] Patrick de Mars and Aidan O’Sullivan. “Applying reinforcement learning and tree search to the unit commitment problem”. In: *Applied Energy* 302 (2021), p. 117519.

[183] Qiuhua Huang et al. “Adaptive Power System Emergency Control using Deep Reinforcement Learning”. In: *IEEE Transactions on Smart Grid* (2019).

[184] Wenqi Cui, Jiayi Li, and Baosen Zhang. *Decentralized Safe Reinforcement Learning for Voltage Control*. 2021. arXiv: 2110.01126 [eess.SY].

[185] Juan S Giraldo et al. “A fixed-point current injection power flow for electric distribution systems using Laurent series”. In: *Electric Power Systems Research* 211 (2022), p. 108326.

[186] Raoul Bernards, Johan Morren, and Han Slootweg. “Statistical modelling of load profiles incorporating correlations using Copula”. In: *2017 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe)*. IEEE. 2017, pp. 1–6.

[187] Edgar Mauricio Salazar Duque et al. “Conditional multivariate elliptical copulas to model residential load profiles from smart meter data”. In: *IEEE Transactions on Smart Grid* 12.5 (2021), pp. 4280–4294.

[188] Morsal Salehi and Mohammad Mahdi Rezaei. “An improved probabilistic load flow in distribution networks based on clustering and Point estimate methods”. In: *Energy and AI* 14 (2023), p. 100272.

[189] Xueru Lin et al. “Component modeling and updating method of integrated energy systems based on knowledge distillation”. In: *Energy and AI* 16 (2024), p. 100350.

[190] Thiago Serra, Christian Tjandraatmadja, and Srikumar Ramalingam. “Bounding and counting linear regions of deep neural networks”. In: *International Conference on Machine Learning*. PMLR. 2018, pp. 4558–4566.

CURRICULUM VITÆ

Shengren HOU

EDUCATION

2021–2024	**Technische Universiteit Delft** Ph.D. Candidate in Electrical Engineering and Computer Science
2018–2021	**Guang Xi University, Guangxi, China** M.Sc. in Electrical Engineering
2014–2018	**Northeast Electric Power University, Jilin, China** Bachelor of Technology in Electrical and Electronics Engineering

AWARDS AND ACHIEVEMENTS

Best Project Award, China Reinforcement Learning Application Competition 2022

- Reinforcement Learning Innovation and Creativity Competition held by Shanghai Digital Brain Laboratory, POLIXIR, and Jiangsu Association of Artificial Intelligence.

National Excellent Graduate Student Award, Education Ministry of the People's Republic of China, 2020
1st **Prize Academic Scholarship**, Guangxi University (2019, 2020)

POSITIONS OF RESPONSIBILITY

Board Member of the Association, Vereniging voor Chinese Wetenschappers en Ingenieurs in Nederland

- Collaborate to organize the entrepreneurship and social activities for Chinese engineers in the Netherlands.

List of Publications

First Author Publications

- **S. Hou**, Palensky P, Vergara P P. Safe Imitation Learning-based Optimal Energy Storage Systems Dispatch in Distribution Networks[J]. arXiv preprint arXiv:2411.00995, 2024.
- **S. Hou**, E. M. Salazar Duque, P. Palensky, Q. Chen, and P. P. Vergara, *A Mix-Integer Programming Based Deep Reinforcement Learning Framework for Optimal Dispatch of Energy Storage System in Distribution Networks*, Journal of Modern Power Systems and Clean Energy, 2024.
- **S. Hou**, A. Fu, E. M. Salazar Duque, P. Palensky, Q. Chen, and P. P. Vergara, *DistFlow Safe Reinforcement Learning Algorithm for Voltage Magnitude Regulation in Distribution Networks*, Journal of Modern Power Systems and Clean Energy, 2024.
- **S. Hou**, S. Gao, W. Xia, E. M. Salazar Duque, P. Palensky, and P. P. Vergara, *RL-ADN: A High-Performance Deep Reinforcement Learning Environment for Optimal Energy Storage Systems Dispatch in Active Distribution Networks*, arXiv preprint arXiv:2408.03685, 2024.
- **S. Hou**, E. M. Salazar Duque, P. Palensky, and P. P. Vergara, *Optimal Energy System Scheduling Using a Constraint-Aware Reinforcement Learning Algorithm*, International Journal of Electrical Power & Energy Systems, vol. 152, 2023.
- **S. Hou**, E. M. Salazar Duque, P. P. Vergara, and P. Palensky, *Performance Comparison of Deep RL Algorithms for Energy Systems Optimal Scheduling*, 2022 IEEE PES Innovative Smart Grid Technologies Conference Europe (ISGT-Europe), pp. 1-6, doi: 10.1109/ISGT-Europe54678.2022.9960642.

Co-Author Publications

- W. Xia, H. Huang, E. M. Salazar Duque, **S. Hou**, P. Palensky, and P. P. Vergara, *Comparative Assessment of Generative Models for Transformer- and Consumer-Level Load Profiles Generation*, Sustainable Energy, Grids and Networks, vol. 38, 2024, doi: 10.1016/j.segan.2024.101338.

ACKNOWLEDGEMENTS

Throughout my Ph.D. journey, I have been privileged to receive the support, encouragement, and guidance of numerous individuals. I am deeply appreciative of their time, expertise, and resources, all of which have contributed immensely to the successful completion of this thesis. I would like to take this opportunity to express my heartfelt thanks to everyone who has supported me throughout this challenging yet rewarding journey.

First and foremost, I would like to extend my deepest gratitude to Prof. Peter Palensky, my promoter, for his exceptional support, guidance, and expertise throughout my doctoral studies. Peter's commitment to my academic progress, insightful feedback, and unwavering support have been invaluable. I have greatly benefited from his profound knowledge and his ability to steer my research in the right direction.

Beyond his academic mentorship, two personal experiences with Peter have left a lasting impression on me. During a particularly difficult period when I faced financial challenges, I reached out to him for advice. Despite it being outside his formal responsibilities, Peter went above and beyond to support me, helping me find part-time academic projects that not only supplemented my income but were also closely aligned with my Ph.D. research. When I told him he was not obligated to help in this way, he simply said, "As a manager, part of my job is to ensure my team can work without distractions." This gesture deeply moved me and cemented my respect for him as both a leader and a mentor.

Another memorable experience came during the later stage of my Ph.D. journey when I began to realize that I wanted to transition into the industry. I aimed to integrate my smart decision agent more closely with real-world electricity markets. Peter gave me tremendous flexibility and support, encouraging me to explore opportunities in electricity market trading. This support allowed me to search for internships and jobs in the field, laying the foundation for my future career.

I am also sincerely grateful to Dr. Pedro Vergara, my copromotor and daily supervisor, whose thoughtful guidance and invaluable feedback have been crucial in shaping this work. Pedro's dedication to revising my work, providing insightful suggestions for experiments, and encouraging me to think critically has significantly contributed to my research development. As Pedro's first Ph.D. student, I have benefited enormously from his meticulous and hands-on mentorship. He has guided me through every step of the research process, from conducting comparative experiments to writing papers, presenting research, and even learning how to craft LaTeX formulas more efficiently.

One vivid memory of our collaboration is our very first online meeting, where Pedro brought my research proposal and invited his friend Mauricio, an expert in machine learning and power systems, to provide additional insights. Their valuable input during that session was both touching and inspiring. Over time, Pedro has become not only a supervisor but also a dear friend. I have witnessed his growth from a fresh co-promoter into an excellent educator and principal investigator. I recall our time at the ISGT 2022

conference, where I suggested to Pedro that he should network more. His candid response was, "Shengren, I don't know how to social either; I guess we'll both have to learn on the go."

My deepest appreciation goes to my family for their unwavering support throughout this journey. I am particularly grateful to my parents (Dianlei Hou and Hongwei Li), for their endless love, patience, and belief in my abilities. Their encouragement during challenging times has been my constant source of strength and motivation. I would also like to extend my heartfelt gratitude to my girlfriend, Ping Mao, for her constant support, patience, and understanding. Your belief in me and the sacrifices you made for our shared dreams have been the bedrock of my motivation. I am immensely grateful for your unwavering encouragement and for standing by my side throughout this journey. Special thanks to my father's cousin, Xuhua Hou, for his continuous support and guidance since my university days. I have learned so much from you, and your influence has been invaluable to me.

I am deeply thankful to my coauthors and collaborators, Prof. Qixin Chen, Dr. Aihui Fu, Mrs. Xue Yao, Dr. Mauricio Salazar Duque, for their invaluable contributions and insights throughout our joint research efforts. Working together has significantly enriched the quality of this thesis, and I am grateful for their expertise and collaboration.

I am also thankful to all my colleagues and friends at IEPG. Sepcial thanks to Prof. Marjan, Dr. Chenguang Wang, Dr. Yigu Liu, Dr. Le Liu, Dr. Na Li, Dr. Ye Ji, Dr. Ties, Dr. Haixiao, Dr. Ajay, Dr. Digvijay Gusain, Ali, Mert, Kutay, Neda, Benovia, Runyao Yu, Weijie Xia, Nan Lin, Shuyi Gao, Haiwei Xie, Lanting Zeng, Chuyi Li, Zhisheng Xiong, Shen Yan, Zeynab, Demitries, Wouter, Stavros, Lily Li, for the stimulating discussions, insightful exchanges of ideas, and the enjoyable moments we shared. Thanks to my office mates Nanda, Rohan, Runqi. The support of this diverse group have been instrumental in broadening my horizons and enhancing my research experience. Special thanks to Mrs. Carla, Mr. Remko, whose assistance with technical support, administrative matters contributed to the smooth progression of my research.

I would also like to express my gratitude to my friends including Qianyi Chen, Ran Zhu, Mingkun Yang, Yang Yang, Sifeng He, Kai Liu, Xuan Liu, Xiujie Shan, Yuexiang Chen, Bowen Li, Qing Yong, Yang Jin, Yiheng Zhang, Runzhang Hong, Ting Hu, Erqian Tang, Huaizhi Yang (Leo), Minghe WU, Yaoguan Yue, Peng Liu, Hao Liu, Shuang Li, Ximing Li, Liang Yue, Zijing Cui, Saige Wang, Enze Zhang, Jinheng Li, Eric Cao. Our weekends together have been full of enjoyable moments. Sepecial thanks to Chris Green and Remy Sun for their help and support in IELTS learning.

I would like to express my thanks to colleagues in VCWI: Xiyu Ouyang, Zixian Bao, Yingying Dou, Wei Zhou, Yang Qiu, Zhenyu Gao, Yuwei Wang, Frank Wang, Xiangmin Weng, Peng Liu, Chenyu Zhou, Meixiu Tan, Chenhui Chang, Jianyao Jin and many others, for their friendship and uplifting conversations that have brought balance and joy to my life throughout my Ph.D. journey. Special thanks to my leader and colleagues in NorthPool: Roald, Margot, Susan, Dies, Tiemen, Teun, Len, Cas, Cheng Liu who supported my research and work. I also would like to express my heartfelt gratitude to Professor Hanbo Zheng for his guidance in both my life and career. He has been not only a mentor in my academic journey but also a close friend in my personal life.

I would like to express my heartfelt gratitude to my homeland, China, for providing

funding that supported both my research and my life during my PhD journey. Coming from an ordinary family, my parents' income could not afford the cost of living in Europe. At the time, Peter did not have funding for a supported PhD position. Without the generous support of the China Scholarship Council (CSC), I would not have had the opportunity to conduct research at IEPG. I am fully aware that this funding comes from the taxes paid by the Chinese people. It is their contribution that made my education and research possible. I feel a deep sense of responsibility and appreciation, and I will strive to give back through meaningful work and continued contribution to society. I am also deeply grateful to the Netherlands for the excellent research environment and the institutional support that made it possible for me to grow as a researcher. I will always remember and appreciate all the support I have received throughout this journey. I will keep remember and appreciate that all these supports.

To all those who have contributed to this work in ways big and small, directly or indirectly, I extend my deepest gratitude. This Ph.D. journey would not have been possible without your support, and I am truly thankful for making this experience both memorable and transformative.

In the end, I want to say something to myself in case of losing it in the future:

- **Stay courageous**; courage is a rarer quality than intelligence, so don't lose it.
- **Stay indignant**; do not turn cynical about the injustices and vast inequalities in society, nor should you bury your head in the sand. Use your influence to inspire others to work together towards positive change.
- **Stay humble**; remember that what you have achieved is not solely due to your intelligence and hard work. Many people work just as hard but do not have the same resources or opportunities.

APPENDIX

A. PROOF OF MIP-DQN

A sketch of a mathematical proof that ensures that the proposed MIP-DQN model provides the optimal solution while strictly enforcing linear constraints is presented below. To do this, we first assume the feasibility to the problem presented in Sec. 2.2 and also present (and adapt notation to match this paper) the Corollary 19, from [190] as,

Corollary 19: If the input (s,a) of the Q-network is a polytope and the DNN is a rectifier network (i.e., ReLU activation functions are used), then the mapping from input (s,a) to the output $Q(s,a)$ of such a Q-network is mixed-integer representable.

The proof of Corollary 19 is available in [190]. Note that this corollary implies that for any rectifier DNN, a mixed-integer formulation exists as long as the input is bounded. The Q-network used in the proposed MIP-DQN algorithm is a DNN with a rectifier activation function while the input (s,a) are bounded as these correspond to the state and action variables as presented in Sec. 2.3. We denote the optimal solution to this MIP formulation as (s^*,a^*) whose optimal objective function value is $Q(s^*,a^*)$.

Now, the extended MIP formulation obtained by adding on top of the MIP representation of the $Q(s,a)$ an equality constraint (in this case, (2.4)) is also a feasible MIP representation. This is a consequence of the fact that such a mixed-integer representation of $Q(s,a)$ is composed of a set of linear regions whose unions form a bounded polyhedron (or polytope) (see Theorem 20 in [190]), which we denote this here as $\mathscr{S}$ (see a representation in Fig 2.4). The addition of (2.4) to $\mathscr{S}$, which is also a linear constraint, does not modify its nature of a bounded polyhedron (or polytope).

By exhaustion, two cases are distinguished: In the first case, the extended bounded polyhedron $\mathscr{S}^{'} = \mathscr{S} \cup (2.4)$ is empty, rendering the solution of the MIP unfeasible, i.e., equality constraint in (2.4) cannot be met. This is not possible as we assumed feasibility for the optimization problem. In the second case, $\mathscr{S}^{'}$ is not empty, in which an optimal solution exits and is feasible. If this is the case, and denoting such optimal solution as $(s^{'},a^{'})$, such solution meets the following condition: $Q(s^{'},a^{'}) \leq Q(s^*,a^*)$. This condition simply implies that $(s^{'},a^{'})$, by meeting the equality constraint in (2.4), will at least have a q-value that is in the limit the same as the optimal solution $Q(s^*,a^*)$. This proves the fact that by solving the extended MIP formulation, a feasible and optimal solution that meets the equality constraint (2.4) is obtained. Nevertheless, it is important to highlight that optimality here relates to the good quality solution provided by the trained Q-network.

B. WORKFLOWS FOR MODULES IN RL-ADN

B.1. DATA MANAGER WORKFLOW

`GeneralPowerDataManager` modular, is a unified data manager. Designed for automation, this class standardizes various data preprocessing tasks, as follows:

- Loads time-indexed data directly from standard CSV files.
- Classifies columns pertaining to active and reactive power, renewable energy generation, and electricity pricing, autonomously.
- Cleans and checks the data, filling in missing values, ensuring data continuity and integrity.
- Segregates the dataset into distinct training and test sets based on temporal delineation.
- Offers utility methods, such as `select-timeslot-data` and `select-day-data`, enabling precise data extraction tailored to the RL training needs.

When the `GeneralPowerDataManager` class is initialized, it undergoes a series of operations: it verifies the data's integrity, replaces any NaN values, and partitions the dataset into training and testing parts as required. These preliminary tasks ensure that data quality is maintained and provide ease of access and utilization for subsequent RL training processes.

B.2. DATA AUGMENTATION WORKFLOW

The augmentation process involves several sophisticated statistical techniques, outlined as follows:

- The `ActivePowerDataManager` class, a subclass of the `GeneralPowerDataManager`, preprocesses the input data, fills missing values through interpolation, and restructures the data into an appropriate format for augmentation.
- A Gaussian Mixture Model (GMM) is fitted to the marginal distribution of historical active power data for each node and time step, capturing the underlying distribution of power consumption.
- The Bayesian Information Criterion (BIC) is employed to select the optimal number of components for each GMM, ensuring that the model complexity is balanced against the goodness of fit.
- A Copula-based approach is then applied, which models the dependency structure between different nodes and time steps, allowing for the generation of synthetic data points that maintain the correlation observed in historical data.
- The `augment_data` method leverages the GMM and Copula to produce new data samples, which are then transformed from the probabilistic space back to the power data scale.

The `TimeSeriesDataAugmentor` modular interacts with the data manager to retrieve the necessary preprocessed data, and then applies its augmentation algorithms to produce an augmented dataset. The output is a synthetic yet realistic dataset that reflects the variability and unpredictability inherent in power systems. This enriched dataset is crucial for training RL agents, providing them with a diverse range of scenarios

to learn from and ultimately resulting in a more adaptable and robust decision-making policy.

Upon completion of the augmentation process, the synthetic data is saved to a CSV file, facilitating easy integration into the training pipeline. This automated and sophisticated data augmentation procedure enhances the RL-ADN framework's capability to train more effective and resilient RL agents for the distribution network ESSs operations.

Deep RL for Optimal Distribution Networks Operation

Shengren Hou

