MicroPPO: Safe Power Flow Management in Decentralized Micro-Grids with Proximal Policy Optimization

Daniel Ebi, Edouard Fouché, Marco Heyden and Klemens Böhm
Karlsruhe Institute of Technology
Karlsruhe, Germany
{daniel.ebi, edouard.fouche, marco.heyden, klemens.boehm}@kit.edu

Abstract

Future sustainable energy systems require the integration of local renewable energy sources (RES) into decentralized micro-grids, each containing RES, energy storage systems, and local loads. A substantial challenge associated with micro-grids is the optimization of energy flows to minimize operating costs. This is particularly complex due to (a) the fluctuating power generation of RES, (b) the variability of local loads, and (c) the possibility of energy trade between a micro-grid and a larger 'utility grid' that it connects to. Existing methods struggle to manage these sources of uncertainty effectively. To address this, we propose MicroPPO, a reinforcement learning approach for real-time management of power flows in such small-scale energy systems. MicroPPO introduces a novel definition of the environment as a Markov Decision Process (MDP) with a continuous and multi-dimensional action space. This enables more precise control of power flows compared to discrete methods. Additionally, MicroPPO employs an innovative actor network architecture featuring multiple network branches to reflect the individual action dimensions. It further integrates a differentiable projection layer that enforces the feasibility of actions. We assess the performance of our approach against state-of-the-art methods using real-world data. Our results demonstrate MicroPPO's superior convergence towards near-optimal policies.

This file corresponds to the accepted version of the manuscript published in *2024 IEEE 11th International Conference on Data Science and Advanced Analytics (DSAA), San Diego, CA, USA, 2024, DOI: 10.1109/DSAA61799.2024.10722774.*

1 Introduction

Renewable energy sources (RES), such as photovoltaic (PV) panels, play a crucial role in the transition to sustainable energy systems. This introduces new challenges related to the decentralized control of locally interconnected RES, storage units, and loads [1]. In private households or within companies, the interconnection of such components forms a small-scale energy system, which we call a decentralized micro-grid.

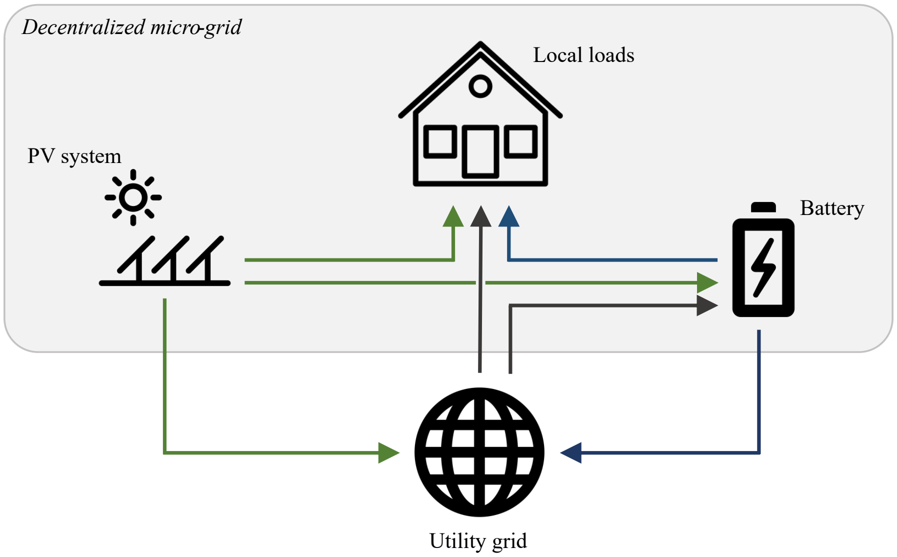

Figure 1: Our micro-grid model consists of a PV system, a battery, and local loads. Moreover, the micro-grid can connect to a utility grid.

Figure 1 shows a decentralized micro-grid that consists of a photovoltaic (PV) system, a battery, and local loads. It can operate autonomously in an ‘island mode’ or connect to a larger ‘utility grid’, enabling energy exchanges. The arrows in the figure indicate the possible directions of power flows.

Managing these power flows is particularly relevant since the number of installed PV panels on private roofs has significantly increased in recent years [2]. PV systems are frequently accompanied by a battery to achieve a certain degree of energy autonomy, so our model reflects a common, state-of-the-art architecture for private micro-grids.

Effective management of power flows within decentralized micro-grids leads to a decrease in operating costs and an increase in renewable energy consumption. However, the uncertain nature of RES, coupled with dynamic load patterns and fluctuating energy prices, make supply and demand extremely hard to predict [3]. Furthermore, micro-grid operators must quickly make decisions to buy or sell the exact amount of power they consume or produce for each market period, e.g., on an hourly basis. Thereby, leveraging energy storage systems (ESS) enables operators to optimize the distribution of energy based on its availability and costs. In other words, such decisions must be automated, using intelligent energy management systems (EMS).

Existing approaches, typically based on mixed-integer linear programming [4] or dynamic programming [5], are computationally expensive and lack flexibility. Additionally, they can only optimize in an offline manner, meaning they operate in hindsight or require reliable forecasts. To address these limitations and enable real-time execution, the optimal power flow management problem is often formalized as a sequential decision-making problem. Model-based approaches such as rolling horizon optimization [6] demand extensive domain knowledge and a precise parametric definition of uncertainties, which might not be feasible in practical applications. Hence, recent reinforcement learning (RL) based approaches allow for learning a policy without the need for a pre-defined model.

The setting we address (cf. Figure 1) involves a multi-component environment with a multi-dimensional action space. It comprises continuous, interconnected actions; even a slight variation in one dimension can result in an infeasible action. Literature [7] suggests that existing RL-based approaches typically struggle with large state-action spaces. Hence, such approaches tend to be limited to discrete action spaces.

To overcome these weaknesses, we propose MicroPPO, a new method for real-time power flow management in decentralized micro-grids that can cope with continuous, interdependent action spaces. It leverages Proximal Policy Optimization (PPO) [8], a powerful model-free deep reinforcement learning (DRL) method.

Our **contributions** are as follows:

- We model the optimal power flow management problem in a decentralized micro-grid as a Markov Decision Process (MDP). We introduce a continuous, multi-dimensional action space that allows the agent to dynamically control the PV system, battery, and energy exchange with the utility grid.
- We present an actor network architecture that enables the agent to learn a near-optimal policy in domains characterized by multi-dimensional action spaces, particularly in the presence of external constraints. Our approach encodes a latent representation of the current state and coordinates different network branches that reflect the individual action dimensions. To enforce the constraints of physical systems, MicroPPO incorporates a differentiable projection layer that maps the selected action to the closest action in the space of feasible actions.
- We present a data generator that augments household energy consumption data with prices from the wholesale electricity market and renewable energy generation data. The generator can serve as a new benchmark for evaluating EMS for decentralized micro-grids.
- We use the generator to compare MicroPPO against mixed-integer linear programming (MILP) methods, rule-based approaches, and state-of-the-art DRL-based EMS.

Our novel algorithm demonstrates superior performance compared to other DRL algorithms. With its continuous action space, MicroPPO offers more fine-grained control of power flows compared to discrete methods. In particular, it achieves a lower error compared to the upper bound obtained by solving the MILP formulation with perfect forecast information.

2 Related Work

Most related work falls into one of two categories: model-based and model-free techniques.

2.1 Model-based approaches

Traditional approaches to micro-grid energy management tend to be based on a model that captures the dynamics of the system. Rule-based approaches [9, 10, 11] use domain knowledge to ensure system constraints are always satisfied. However, applying a pre-defined set of rules is generally far from optimal.

Techniques such as (non-)linear programming [12], mixed-integer linear programming (MILP) [4, 13], and dynamic programming [5, 14] solve the problem via offline optimization. These methods can provide optimality guarantees but scale poorly with the dimensionality of the problem instances [15]. Robust optimization [16] and stochastic optimization [17] alleviate this issue to some extent. Rolling horizon optimization, also known as model predictive control (MPC), uses an additional uncertainty estimator and allows for real-time operation of energy systems [18, 19]. However, all these approaches require an explicit representation of the micro-grid, limiting scalability and flexibility [20]. Whenever the distribution of uncertainty changes, one needs to redesign the model, predictor, and solver components [7].

2.2 Model-free approaches

To break away from the reliance on a fixed model, more recent approaches tend to build on reinforcement learning (RL). By interacting with the environment, RL techniques learn a so-called policy that serves as a near-optimal strategy [21].

One well-known RL technique is Q-Learning, which has been applied to energy management [22, 23]. However, Q-Learning tends to struggle in micro-grid contexts due to the very large state-action spaces involved [7]. Hence, deep reinforcement learning (DRL) techniques have been developed, which can manage larger state-action spaces using deep neural networks. In general, DRL-based EMS differ in how much control they have over micro-grid components. We distinguish between two types:

- The EMS can control individual components of the micro-grid. Various approaches are based on Deep Q-Networks (DQN) [24, 25] and double DQN [26]. For example, [24] applies a DQN to both maximize the profit of RES operators and reduce revenue variability using an ESS.

- The EMS jointly controls several micro-grid components. It uses more complex objective functions and composed action spaces to do so. For example, [7] and [27] apply a DQN to control battery operations and transactions with the utility grid. In [28], the authors explore different value-based and policy-based DRL algorithms for energy management in micro-grids with flexible demand. Moreover, they present novel adaptations of Proximal Policy Optimization (PPO) and the A3C algorithm that integrate an experience buffer to enhance data efficiency.

To our knowledge, most existing work on DRL-based EMS in micro-grids considers a discrete action space. As dealing with large state-action spaces is difficult, discretizations tend to be coarse, which limits granularity in micro-grid operations.

To ensure the reliability and safety of micro-grid operations at all times, it is crucial to enforce the agent's adherence to system constraints. Existing work tends to deal with such constraints by implementing soft penalties for infeasible actions. However, such penalties may incentivize feasibility, they do not enforce the agent's adherence to system constraints. For instance, in [24], the authors add a penalty term to the reward function to prevent the agent from choosing infeasible battery actions. [29] distinguishes different methods for enforcing hard constraints in RL settings: action replacement, action masking, and action projection. Existing DRL-based approaches for power flow management in micro-grids that operate on discrete action spaces, such as [7], use action masking. I.e., the agent can only choose from actions in the safe action space. However, action masking is hard to implement for continuous action spaces. Hence, [30] proposes representing the underlying deep neural network as a MILP. Alternatively, [31] ensures feasibility by incorporating a differentiable projection layer into the policy in an inverter control scenario. One advantage is that the policy is still updated via gradient propagation through that layer.

Our approach, MicroPPO, is model-free and is the first to apply PPO together with continuous action space modeling in the context of power flow management in decentralized micro-grids. PPO is particularly well suited for this application due to its stability, sample efficiency, and ability to handle continuous actions [8]. We combine PPO with a differentiable projection layer [31] to map actions to the space of feasible actions, ensuring the safe operation of the energy system. Unlike other RL-based techniques, MicroPPO offers an MDP that models the continuous action space. Our experiments show that this leads to more precise control of the power flows.

3 Preliminaries

3.1 Micro-grid Model

Our model, illustrated in Figure 1, reflects a common micro-grid architecture. It comprises a photovoltaic (PV) system, a battery, and local loads, and also allows for energy trading with the utility grid.

We assume that the micro-grid operators have full control over the operation of the actionable components of the micro-grid, i.e., they can control the direction and amount of power flow between the components. The goal is to fulfill the load demand while minimizing operating costs. The latter emerge from using the battery and buying power from the utility grid. Revenues may be obtained by selling power to the grid. The following paragraphs introduce the different components of the micro-grid in detail.

Utility Grid. The micro-grid can connect to the utility grid G, allowing buying and selling power in real-time. Let c_t denote the real-time electricity market price at time t. Further, we assume that the costs of purchasing power from the grid exceed the selling prices. Therefore, we model the selling prices as a discount $\beta \in [0, 1]$ of the electricity market price. So the transaction costs of the micro-grid at time t are

$$f_t = c_t \cdot p^G_{in,t} - \beta \cdot c_t \cdot p^G_{out,t}, \tag{1}$$

where $p^G_{in,t} \geq 0$ refers to the power inflow from the utility grid, i.e., power purchased, and $p^G_{out,t} \geq 0$ is the power outflow, i.e., power sold. Given a sufficiently coarse time resolution, $\Delta t = t_{i+1} - t_i$, both $p^G_{in,t} \geq 0$ and $p^G_{out,t} \geq 0$ might be larger than zero. I.e., we purchase power for $x\%$ of the current market period, and for the remaining time, which is $(1 - x) \cdot \Delta t$, we export power to the utility grid.

Local Loads. For simplicity, we aggregate the loads of elements related to the household H that consume electricity, such as lighting devices, heating and cooling systems, or other electrical devices. Let l_t^H be the aggregated load demand at time t. In energy systems, renewable generation and load demand often fluctuate. To ensure safe operation of the micro-grid, one strives to meet the load demand at all times by implementing a reserve management strategy. In our micro-grid model, we can purchase any power deficit from the utility grid G at any given time t to maintain power balance:

$$p_{H,t}^{PV} + p_{H,t}^{B} + p_{H,t}^{G} = l_t^H. \tag{2}$$

$p_{H,t}^{PV}$, $p_{H,t}^{B}$, and $p_{H,t}^{G}$ are the power flows from the PV system, the battery and the utility grid to the household, respectively.

Renewables Generation. The renewable generation system comprises PV panels with a maximum joint capacity U^{PV}. Excess power generated by the PV system during low power demand can either be stored in the battery or sold to the utility grid. Let g_t^{PV} be the PV generation at time t.

Energy Storage System (ESS). The ESS consists of a battery B that is mainly characterized by the maximum capacity U^B, and the charging and discharging efficiencies η_{ch}^B and η_{dis}^B, respectively. At each time step t, the dynamics of the battery are modeled by

$$SoC_t^B = SoC_{t-1}^B + \frac{\Delta t \cdot (p_{in,t-1}^B + p_{out,t-1}^B)}{U^B}, \tag{3}$$

where $SoC_t^B \in [SoC_{min}^B, SoC_{max}^B]$ denotes the state of charge at time t. SoC_{min}^B and SoC_{max}^B are the minimum and maximum allowable energy levels of the battery, respectively.

The actual charging power $p_{in,t}^B \in [0, R_{ch}^B]$ is defined as the proportion of the minimum between the external power available for charging $p_{ext,t}^B$ and the maximum available upward power given the current state of charge, i.e.,

$$p_{in,t}^B = v_t \cdot min\Big(p_{ext,t}^B \cdot \eta_{ch}^B, (SoC_{max}^B - SoC_t^B) \cdot \frac{U^B}{\Delta t}\Big) \tag{4}$$

with $v_t \in [0, 1]$. Note that $p_{in,t}^B$ is constrained by the battery's charging rate limitation R_{ch}^B. The actual discharging power $p_{out,t}^B \in [0, R_{dis}^B]$ is given as

$$p_{out,t}^B = w_t \cdot \Big((SoC_t^B - SoC_{min}^B) \cdot \frac{U^B}{\Delta t}\Big) \tag{5}$$

with $w_t \in [0, 1]$. v_t and w_t are external factors that control the amount of available power used to charge the battery or discharged from the battery, respectively. The micro-grid operator needs to decide on these factors at each time t, with at least one of these variables being 0, so that the battery is not charged and discharged simultaneously.

Both charging and discharging operations are associated with some fixed costs ξ^B per kWh that account for the acquisition costs of the storage system. We compute ξ^B as

$$\xi^B = \frac{C^B}{U^B \cdot \omega^B \cdot (SoC_{max}^B - SoC_{min}^B) \cdot \eta_{dis}^B},$$

where C^B denotes the battery's acquisition costs, and ω^B is the number of maximum charging/discharging cycles. The intuition behind this levelized cost of storage (LCOS) metric is to relate the battery costs to the potential total amount of energy discharged over its lifetime, as represented in the denominator [32].

3.2 Global Optimal Solution

We formulate the power flow management problem using mixed-integer linear programming (MILP) for the decentralized micro-grid described in the previous section. Solving this MILP allows us to obtain the global optimal power flow schedule, which serves as the upper bound in our experiments.

In this MILP formulation, the micro-grid operator must make decisions on the power flows for the entire time horizon T at once, i.e. $t \in \{0, 1, \ldots, T\}$. These decisions rely on information such as the generated power from PV systems, load demand, and the battery's state of charge.

In this work, we adapt the offline optimization problem formulation from [4] to our setting, resulting in the optimization problem described in Eqs. 6a-6n. The operator decides on 1) the flow of power from the PV system to the household $p^{PV}_{H,t}$, the battery $p^{PV}_{B,t}$ and the utility grid $p^{PV}_{G,t}$, 2) the power flow from the battery to the household $p^{B}_{H,t}$ and the utility grid $p^{B}_{G,t}$, and 3) the amount of power purchased from the utility grid $p^{G}_{B,t}$ to charge the battery or to fulfill the load demand $p^{G}_{H,t}$. The objective function in Eq. 6a aims to maximize revenue over the entire time horizon T, accounting for both transaction costs and the operating costs of the battery. Following [4], we introduce the binary decision variables $y^{B}_{ch,t}$ and $y^{B}_{dis,t}$ to guarantee that the battery is never charged and discharged simultaneously.

$$\max \sum_{t=0}^{T} \Big(- f_t - \xi^B \cdot \big(p^{B}_{H,t} + p^{B}_{G,t} + p^{PV}_{B,t} + p^{G}_{B,t}\big)\Big) \tag{6a}$$

s.t.

$$p^{PV}_{H,t} + p^{B}_{H,t} \cdot \eta^{B}_{dis} + p^{G}_{H,t} = l^{H}_{t} \quad \forall t, \tag{6b}$$

$$p^{PV}_{H,t} + p^{PV}_{B,t} + p^{PV}_{G,t} = g^{PV}_{t} \quad \forall t, \tag{6c}$$

$$p^{PV}_{B,t} + p^{G}_{B,t} \leq R^{B}_{ch} \cdot y^{B}_{ch,t} \quad \forall t, \tag{6d}$$

$$p^{PV}_{B,t} + p^{G}_{B,t} \leq \big(SoC^{B}_{max} - SoC^{B}_{t}\big) \cdot \frac{U^B}{\Delta t} \quad \forall t, \tag{6e}$$

$$p^{B}_{H,t} + p^{B}_{G,t} \leq R^{B}_{dis} \cdot y^{B}_{dis,t} \quad \forall t, \tag{6f}$$

$$p^{B}_{H,t} + p^{B}_{G,t} \leq \big(SoC^{B}_{t} - SoC^{B}_{min}\big) \cdot \frac{U^B}{\Delta t} \quad \forall t, \tag{6g}$$

$$y^{B}_{ch,t} + y^{B}_{dis,t} \leq 1 \quad \forall t, \tag{6h}$$

$$p^{G}_{H,t} + p^{G}_{B,t} =: p^{G}_{in,t} \quad \forall t, \tag{6i}$$

$$p^{PV}_{G,t} + p^{B}_{G,t} =: p^{G}_{out,t} \quad \forall t, \tag{6j}$$

$$p^{PV}_{\alpha,t} \in \mathbb{R}_{\geq 0} \quad \forall \alpha \in \{H, B, G\}, \quad \forall t, \tag{6k}$$

$$p^{B}_{\alpha,t} \in \mathbb{R}_{\geq 0} \quad \forall \alpha \in \{H, G\}, \quad \forall t, \tag{6l}$$

$$p^{G}_{\alpha,t} \in \mathbb{R}_{\geq 0} \quad \forall \alpha \in \{H, B\}, \quad \forall t, \tag{6m}$$

$$y^{B}_{\alpha,t} \in \{0, 1\} \quad \forall \alpha \in \{ch, dis\}, \quad \forall t. \tag{6n}$$

Eq. 6b is the power balance constraint. Eq. 6c sets renewable generation power limits. The ESS is characterized by Eqs. 6d-6g. Specifically, Eqs. 6d-6e establish the battery's charging power limit, while Eqs. 6f-6g define the discharging power limit. Eq. 6h prevents the battery from being charged and discharged simultaneously. Eqs. 6i-6j relate to the power in- and outflow from and to the utility grid, respectively. Lastly, Eqs. 6k-6n define the domain of the decision variables.

Note that either full knowledge of future information, i.e., about renewable generation, load demands and energy prices, or corresponding forecasts are required to solve this MILP formulation. In other words, this technique requires information that is hardly available in practical scenarios or introduces further uncertainty.

Hence, in practice, the real-time management of power flows is typically a sequential decision-making problem. To account for this sequential nature, we can derive a rolling horizon optimization formulation, enabling real-time decisions by sequentially solving the presented MILP for each time step individually. This allows us to compute the information needed for the decision in the current step based on the optimal solution of the previous step. Specifically, we can update the battery's SoC using Eq. 3.

We use both the offline MILP optimization and the adopted sequential optimization as baselines in our experiments.

4 Our Approach

Our model-free approach, MicroPPO, is based on the RL paradigm, where an agent learns optimal control policies through direct interactions with its environment. In the optimal power flow management problem, the agent is the micro-grid operator, and the environment corresponds to our micro-grid model presented in Section 3.1. The objective is to learn a policy that maximizes the expected returns over time.

MicroPPO consists of two main parts: first, the formalization of the sequential decision-making problem as Markov Decision Process (MDP); second, the algorithm to learn a (near-)optimal policy for solving this MDP.

4.1 Formalization as Markov Decision Process

An MDP is represented by a 5-tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{P}, R, \gamma \rangle$, where $\mathcal{S}$ is the set of states, $\mathcal{A}$ is the set of actions, $\mathcal{P}$ is the state transition probability function, R is the reward function, and γ is a discount factor.

Our MDP for the power flow management problem incorporates the assumptions outlined in our micro-grid model in Section 3.1. The near-optimal results of MicroPPO on real-world data suggest that our approximation using an MDP is indeed a useful model for practical applications.

State space $\mathcal{S}$. The state is the information that the agent uses for decision-making at each time step t. We define the state $s_t \in \mathcal{S}$ as $s_t = \Big(m_t, h_t, d_t, l_t^H, g_t^{PV}, c_t, SoC_t^B\Big)$. The variables m_t, h_t, and d_t denote the current month, the current hour, and whether it is a working day, respectively. Note that the load demand l_t^H, renewable generation g_t^{PV}, and the energy market price c_t are forecasts. SoC_t^B is the battery's state of charge at the beginning of period t, obtained from Eq. 3.

Action space $\mathcal{A}$. Given the state s_t, the agent interacts with the environment through the control mechanisms outlined in Section 3.1. The action space comprises two components: (1) the renewable generation and self-consumption action space A^{SC}, and (2) the battery-grid action space A^{BG}. Thereby, $A^{SC} \in [0, 1] \times [-1, 1]$ is a 2-dimensional continuous action space that corresponds to the share of power generated that flows from the PV system to the household or battery, respectively. Negative values in the second dimension denote the share of power stored in the battery that flows to the household. The battery-grid action space $A^{BG} \in [-1, 1]$ has one dimension that specifies the extent of the battery being charged with power from grid for non-negative values, or discharged to the grid otherwise. Hence, we define the action $a_t \in \mathcal{A} = A^{SC} \times A^{BG}$ as

$$a_t = \Big(a_t^{SC_1}, a_t^{SC_2}, a_t^{BG}\Big).$$

Following the assumptions made in our micro-grid model, we automatically purchase any power deficit from the utility grid or sell over-produced power to the utility grid. However, $\mathcal{A}$ is a constrained action space as some actions might lead to infeasible states due to physical constraints. We define the space of feasible actions for time step t as Γ_t using a set of constraints:

$$\Gamma_t = \left\{ \tilde{a}_t \;\middle|\; \begin{array}{l} 0 \leq \tilde{a}_t^{SC_1} + o_t^B \cdot \tilde{a}_t^{SC_2} \leq 1, \\ o_t^B - 1 \leq \tilde{a}_t^{SC_2} + \tilde{a}_t^{BG} \leq o_t^B \end{array} \right\}, \tag{7}$$

where $o_t^B \in \{0, 1\}$ is an additional binary decision variable that denotes whether the battery should be charged ($o_t^B = 1$) or discharged ($o_t^B = 0$). The constraints provide guarantee that the power taken from the PV system does not exceed its generation. Furthermore, they prevent the battery from being overcharged or extracting more power than is allowed. Simultaneous charging and discharging of the battery is not possible either. To guarantee that the agent only selects actions from Γ_t, MicroPPO implements a so-called differentiable projection layer. We describe the details in Section 4.2.

Transition Probabilities $\mathcal{P}$. Given the state $s_t = s$ and the action $a_t = a$ at time step t, the next state of the micro-grid changes to $s_{t+1} = s'$ with a conditional probability $\mathcal{P}_{ss'}^a$. The transition probabilities are influenced by the uncertainty within the micro-grid. However, as our approach is model-free, we do not need to model transition probabilities explicitly.

Rewards R. To guide learning, the RL agent obtains a numerical signal, the so-called reward r_t, after performing an action and moving to the next state. That reward quantifies the goodness of the action taken. In line with the presented MILP formulation, the agent's objective is to maximize the profit gained from selling energy to the utility grid while minimizing both the costs of power purchased and storage operation costs. To do so, we define the reward function as follows:

$$R_t(s_t, a_t) = r_t = -f_t - \xi^B \cdot (p^B_{out,t} + p^B_{in,t}), \tag{8}$$

where f_t denotes the transaction costs of the micro-grid (cf. Eq. 1), and $\xi^B \cdot (p^B_{out,t} + p^B_{in,t})$ defines the operating costs of the battery given its power in- and outflow at time step t.

4.2 Learning Algorithm

MicroPPO leverages Proximal Policy Optimization (PPO) to learn a (near-)optimal policy for solving the presented MDP. PPO is a policy-gradient DRL algorithm that has been proposed in [8]. It adopts the so-called actor-critic approach.

Actor-critic methods use two neural networks to assist the policy update by considering value function information:

- The critic network, or simply critic, estimates the state value function $V^{\pi_k}(s_t)$ given the current policy $\pi_k = \pi(a_t \mid s_t, \theta_k)$. It is trained based on past experiences.
- The second network, referred to as the actor network, updates the parameters θ_k of policy π_k in the direction suggested by the advantage function $A(a_t, s_t)$. Hereby, the advantage function specifies the advantage of choosing action a_t in state s_t over the expected value of s_t. We can approximate it as $\hat{A}_t$ by using the estimated state value from the critic.

In PPO, policy updates are constrained within a trust region to prevent moving too far from the policy under which the training data was collected. This helps to mitigate the risk of destructive large policy updates.

PPO tries to maximize the clipped surrogate objective function for the actor network,

$$L^c_\pi(\theta_k) = min\Big(\rho_t(\theta_k)\hat{A}_t, clip(\rho_t(\theta_k), 1-\epsilon_t, 1+\epsilon_t)\hat{A}_t\Big),$$

with $\rho_t(\theta_k) = \frac{\pi_k}{\pi_{k-1}}$, and ϵ is a hyperparameter that defines the trust region. The *clip*-operator restricts the probability ratio $\rho_t(\theta_k)$ to $[1-\epsilon, 1+\epsilon]$. The joint objective function combines the actor's objective and the critic's loss. It is a pessimistic bound on the unclipped objective:

$$L^{PPO}_\pi(\theta_k) = \hat{\mathbb{E}}\big[L^c_\pi(\theta_k) - \lambda_v L^v_\pi(\theta_k)\big]. \tag{9}$$

I.e., a change in ρ_t is only included if it worsens the objective, and is discarded if it improves it. The hyperparameter λ_v determines the degree of influence that the critic's loss L^v_π has on the parameter update.

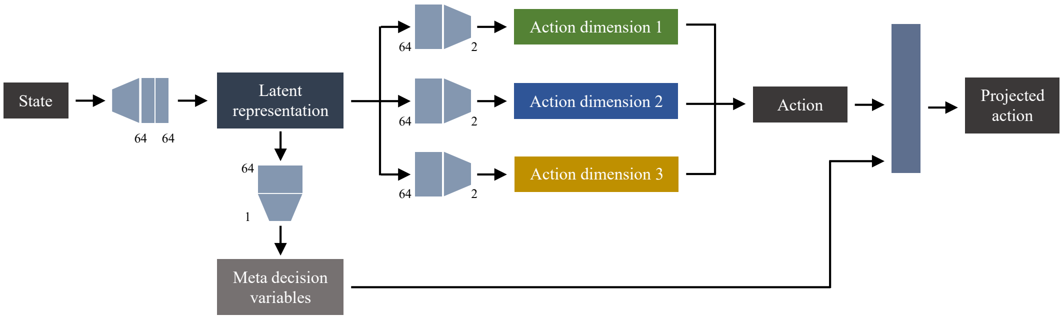

Figure 2: The architecture of the actor network implemented for the MicroPPO agent features a branched structure. The fully connected neural layers are represented by trapezoids in grayish-blue, and the size of each layer (i.e., the number of units) is indicated.

In MicroPPO, the critic network consists of two hidden layers with 32 units each and the ReLU activation function, followed by an output layer consisting of a single linear unit to estimate the state value $V^{\pi_\theta}(s_t)$. We set λ_v to 0.5, ϵ to 0.2 and the discount factor γ to 0.99. We use the Adam optimizer with a learning rate of 0.0085.

Actor Network. Our intuition tells us that, in multi-dimensional action spaces, there may be dependencies between action choices across different dimensions. However, the findings of [33] suggest that selecting actions from multi-dimensional action spaces with a certain degree of independence can lead to better decisions. Following [33], we therefore propose an actor network architecture that distributes the action selection across several network branches corresponding to the action dimensions. We denote them as action branches. Figure 2 provides a conceptual illustration of our actor network architecture.

All action branches are headed by a shared module that encodes a latent representation of the common input state. After selecting an action in each dimension, we obtain the joint action by concatenating the individual dimensions. Hence, given the branched network structure, the total number of network outputs increases linearly with the action dimensionality, in contrast to the usual combinatorial growth observed in settings with multi-dimensional action spaces.

Another fundamental challenge associated with many DRL methods is that they lack consideration of the physical constraints inherent in the systems in which they operate. I.e., we want MicroPPO to ensure the actions' feasibility for safe operation both during training and online execution. That is not trivial because the action space $\mathcal{A}$ is multi-dimensional, so each action affects multiple components of the environment simultaneously. Put differently, the joint action might violate some system constraints as we select the action in each dimension with some degree of independence.

To address this, our actor network implements a subsequent differentiable projection layer that maps each joint action to the closest action in the space of feasible actions. The projection layer can take additional meta-decision variables as inputs to enable the projection or influence its direction. We determine these meta-decision variables in a separate network branch that is also coordinated by the shared module.

Our shared module consists of two hidden layers with 64 units each and the ReLU activation function. Each action network branch consists of a hidden layer with 64 units followed by the hyperbolic tangent activation function. The branch for outputting the meta-decision variables uses a softmax activation after the last layer.

Differentiable Projection Layer. The projection layer mimics a function that maps the obtained joint action at time t to the closest action in the space of feasible actions Γ_t.

In general, to integrate a function as a layer into a neural network, one must define a forward procedure that maps inputs to outputs, and a backward procedure that allows back-propagation of gradients. Typically, the forward procedure of a projection operation involves solving an optimization problem, e.g., using standard convex optimization solvers [34]. However, in our case, Γ_t is non-convex as it includes non-convex constraints resulting from the non-continuous variable o_t^B. Thus, we must relax the envisioned optimization problem to be able to use standard solvers. We do this by treating o_t^B as a meta-decision variable that we compute using the aforementioned additional network branch. In other words, o_t serves as an external parameter to the projection operation, making the optimization problem linear and convex. For deriving the backward pass, we can then use the implicit function theorem as described in [34].

We define $\mathcal{P}_{\Gamma_t} : \mathcal{A} \times \{0, 1\} \to \Gamma_t$ as a L2-norm projection that maps any action $a_t \in \mathcal{A}$, chosen under π_k, to the closest action $\tilde{a}_t$ in the feasible action space Γ_t, given o_t:

$$\mathcal{P}_{\Gamma_t}(a_t, o_t) = \arg \min_{\tilde{a}_t \in \Gamma_t} \|a_t - \tilde{a}_t\|_2^2. \tag{10}$$

After projecting action a_t to $\tilde{a}_t = \mathcal{P}_{\Gamma_t}(a_t)$, we use the projected action $\tilde{a}_t$ to transition to the next state. Additionally, we calculate the projection loss at time t as

$$L_t^{pr} = -\|a_t - \tilde{a}_t\|_2^2.$$

We add L_{pr} to the joint objective function of PPO (cf. Eq. 9) to be able to train the actor-network end-to-end and to guide the agent towards sampling feasible actions from Γ_t. I.e., our actor network is cognizant of the relevant constraints during the learning process, aiming to enhance the overall model performance. The adjusted joint objective function is a weighted sum of the PPO objective function and the projection loss term, given by

$$L_{\pi}^{PPO'}(\theta_k) = L_{\pi}^{PPO}(\theta_k) + \lambda_{pr} L_t^{pr} \tag{11}$$

with hyperparameter $\lambda_{pr} > 0$.

5 Experiments

In this section, we evaluate our approach using real-world data, and compare our results to the state-of-the-art.

We implement MicroPPO using the `OpenAI Gym` framework [35] and the `Stable Baselines 3` library [36]. Additionally, we use the `cvxpylayers` library [34] for differentiable projection. We release our source code and experiments on GitHub[1] with documentation to ensure reproducibility. We run our experiments on a server with 64 cores at 2.4 GHz and 32 GB RAM per task.

5.1 Benchmark Data

We evaluate MicroPPO on real-world energy consumption data of 200 German households from 2019, first published in [37]. We combine this with energy price data from the same time span [38]. To obtain household-level prices, we apply an affine linear transformation to the German day-ahead wholesale electricity market prices. We set the selling price to one quarter of the buying price, i.e., $\beta = 0.25$. Moreover, we incorporate renewable generation data of a German solar PV system that was simulated in [39] using NASA's MERRA-2 [40]. All data is sampled in hourly resolution. For each of the 200 households, we align the PV system's capacity and the nominal capacity of the battery with their annual electricity consumption. Table 1 shows the parameter values for the micro-grid's components we use in our experiments.

We construct a data set comprising load, renewable generation, and price data from 2019 and segment it week by week. This results in a total of 10,400 non-overlapping samples. Each sample comprises 168 data points (7 days $\times$ 24 hours per day $=$168 hours) with three features.

We conduct a 10-fold cross-validation on the household dimension and apply the holdout set methodology on the temporal dimension. I.e., we first reserve 12 weeks of data (one week per month) as a test set. Then, we successively consider 90% of the households (180) for training the model, and use 10% of the households (20) for testing. Therefore, the training set comprises 7,200 samples, and the test set consists of 240 samples. This way, we ensure that we test our approach against households and time intervals that have never been used for training.

Table 1: Overview of the parameters for the micro-grid components

Component	Parameter	Value
PV system	U^{PV}	1.5 kWP per 1 MWh annual consumption
Battery	U^B	1 kWh per 1 MWh annual consumption
	SoC^B_{min}	10 %
	SoC^B_{max}	90 %
	$SoC^B_{t=0}$	50 %
	$R^B_{\text{ch}}/R^B_{\text{dis}}$	$0.5 \cdot U^B$ kWh
	$\eta_{\text{ch}}/\eta_{\text{dis}}$	90 %
Utility grid	β	25 %

[1] `https://www.github.com/EbiDa/MicroPPO`

5.2 Baselines

We benchmark MicroPPO against the following model-based and model-free baselines:

- **Offline Optimization** (B1) uses the MILP formulation as presented in Section 3.2 to obtain a power flow schedule at the horizon of one week. The optimizer uses the actual seven-day-ahead information, i.e., it assumes perfect forecasting. Hence, compared to other approaches, it has an unfair advantage. We treat the obtained global optimal solution as upper bound.
- **Sequential Optimization** (B2) is the single-step variant of B1. I.e., it optimizes the micro-grid operation for a single step ahead using perfect forecasts. Thus, unlike B1, it can be used in a real-time EMS.
- **Rule-based Operation** (B3) seeks to minimize operation costs by applying simple rules that prioritize self-consumption and efficient battery usage by considering energy prices. Specifically, when more power is generated than consumed, the surplus is handled as follows: If the current energy price is higher than the median historical price, the surplus is sold. Otherwise, the surplus is used to charge the battery up to maximum capacity, and only the remaining excess power is sold. When less power is generated than consumed, the demand is covered by storage as much as possible, and the remaining deficit is covered by purchasing energy. This approach, along with the subsequent baselines B4-B7, uses forecasts of l_t^H, g_t^{PV} and c_t.
- **Self-consumption Pattern** (B4) is a rule-based approach similar to B3, but it seeks to maximize the consumption of locally generated power instead of minimizing operation costs. This reflects a common operating strategy used in practice. I.e., when the generated power exceeds the demand, the surplus is used for charging the battery until it is full, and further surplus is sold. When less power is generated than consumed, the battery covers the deficit until it is empty, and further deficit power is bought from the utility grid.

  We provide the pseudocode of the rule-based baselines B3 and B4 in Appendix A.1.
- **PPO-based EMS on continuous action space** (B5) uses, in line with our approach, PPO to solve the MDP presented in Section 4.1, but without applying any action projection. Instead, it enhances the reward function r_t (cf. Eq. 8) by two soft penalties to guide the agent's learning. The first term L_t^{PV} penalizes infeasible actions w.r.t. the PV system, i.e., when more energy is distributed to household, battery, or utility grid than is generated. The second term L_t^B penalizes infeasible battery-related actions, i.e., discharging the battery more than allowed or over-charging the battery, respectively. The revised reward function for B5 is then

  $$r_t^{B5} = r_t + \lambda_{PV} L_t^{PV} + \lambda_B L_t^B,$$

  with the corresponding penalty factors $\lambda_{PV}, \lambda_B > 0$. For λ_{PV} and λ_B, we tested different values. With setting both hyperparameters to 0, the agent failed to learn how to avoid invalid actions. With $\lambda_B = 50$, it avoids choosing invalid actions but acts too conservatively w.r.t. the battery. We finally set λ_{PV} to 10 and λ_B to 5.
- **DQN-based EMS** (B6) is an adaptation of [24] to our micro-grid setting. We use the same state definition as for our approach. However, since DQNs cannot deal with continuous action spaces, we discretize $\mathcal{A}$. We use the same reward function as in our approach, but without adding any penalty as we can simply apply action masking. I.e., we exclude any infeasible action from $\mathcal{A}^{DQN}$.
- **PPO-based EMS on discrete action space** (B7) applies PPO to optimize the agent's policy on the same discrete action space as B6, i.e., $\mathcal{A}^{DQN}$. Hence, as for B6, we neglect any penalty term in the reward function.

We outline the hyperparameter configurations of the baselines B5-B7 used in the experiments in Appendix A.2.

5.3 Model Performance

We train MicroPPO for 1,209,600 steps (7,200 samples × 168 hours), i.e., 7,200 episodes. We update the policy every seven hours based on experiences gained from 24 samples, by iterating over those samples with a batch size of 168 (24 samples × 7 hours per sample). To modulate the influence of the projection loss L_t^{pr}, we set λ_{pr} to 2 (cf. Eq. 11).

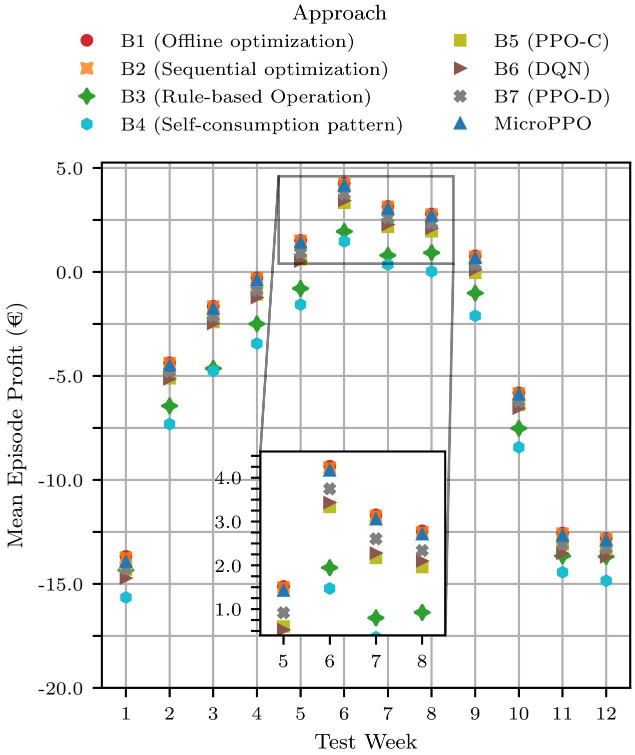

Figure 3: Comparison of the mean episode profit (in €) achieved on the test set by MicroPPO and the different baselines.

We assess the model performance of the different approaches on unseen data using the test set. Figure 3 shows the mean episodic profits achieved by different EMS strategies for the 12 weeks in the holdout set, ordered by month.

Weekly profits. We compare the weekly profits generated using the strategy learned by MicroPPO with those realized by the baselines. MicroPPO achieves only minimally smaller episodic profits than the offline optimization (B1) and the sequential optimization approach (B2), which have the unrealistic advantage of a perfect forecast. Baseline B2 demonstrates near-optimal performance. This confirms our assumption that formulating the optimal power flow management problem as a sequential decision problem in which we base decisions solely on the last state and action is a valid approach.

Using MicroPPO, the operator can achieve a weekly revenue of up to 4.17€ on average in test week 6, while the upper bound for mean episodic profit for this week, obtained with B1, is 4.26€. MicroPPO consistently outperforms the rule-based strategies (B3-B4) and other DRL-based approaches (B5-B7) throughout the entire year, across all test weeks. We can see that it achieves significantly higher profits in the summer weeks (test weeks 6 to 8) on average compared to B5. Although the policy learned with B5 shows higher average profits than those learned with B6 and B7, it fails to maintain action feasibility. B5 handles the system constraints as soft constraints by introducing a penalty term to the reward function. On average, the agent of B5 violates any of the presented system constraints 0.14 times per episode, while MicroPPO and the DRL-based approaches employing action masking (B6-B7) ensure action feasibility at any point in time.

Optimality gap. In addition, we analyze the optimality gap of MicroPPO and the baselines B2-B7 concerning the global optimal solution for operating the micro-grid at minimum costs. In this work, we measure the optimality gap using the relative difference as follows:

$$I(v_i, v_i^{ref}) = -\frac{v_i - v_i^{ref}}{\mid v_i^{ref} \mid}. \tag{12}$$

Note that the closer $I(v_i, v_i^{ref})$ is to 0, the better. We apply this metric to the mean episodic profits, setting v_i^{ref} to the profit obtained by B1 for episode $i \in \{1, 2, ..., 240\}$.

Figure 4 presents the boxplot distributions of the optimality gap concerning the optimal power flow schedule for the various strategies. The median optimality gap for MicroPPO is 0.0234, which is surpassed only by the sequential optimization (B2) with perfect forecasts (median optimality gap of 0.0065). Notably, our approach demonstrates superior performance compared to other PPO-based strategies, namely B5 (median of 0.0466) and B7 (median of 0.0942). Furthermore, in Figure 4, we see that MicroPPO exhibits considerably less variation in the optimality gap than the other DRL-based strategies. Among all the strategies, B2 shows minimal variation around its median optimality gap value.

Our findings underscore the significance of fine-grained action choices, specifically employing a continuous action space, in facilitating the learning of near-optimal policies for cost-effective microgrid operation. The optimality gap distributions of B6 and B7 further support this observation, with both baselines showing higher median optimality gaps compared to the DRL-based approaches using continuous action spaces (B5 and MicroPPO). In particular, the DQN-based approach (B6) seems to have difficulties converging to a near-optimal strategy given the limited number of training episodes. Finally, both rule-based approaches (B3 and B4) fall significantly short of achieving the global optimal solution.

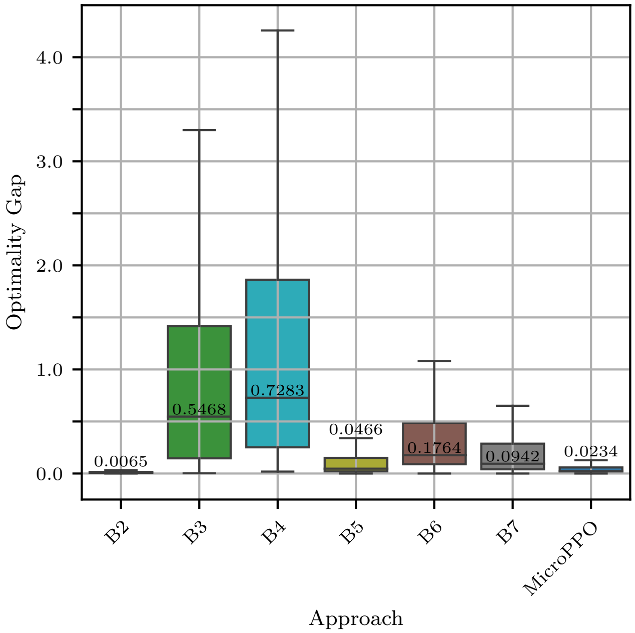

Figure 4: Boxplot distributions of optimality gap on a test set, w.r.t. the global optimal schedule obtained solving the MILP formulation presented in Section 3.2. For each approach, we report its median optimality gap.

6 Discussion

The policy learned by MicroPPO converges to the global optimal solution while guaranteeing safe operation throughout the entire horizon. In contrast, a PPO-based approach that also uses continuous action space but relies on soft constraints to prevent the agent from violating system constraints only encourages the agent to choose actions from the safety set but does not strictly enforce them. This may lead to infeasible or catastrophic system states. However, imposing hard constraints on the agent using a projection layer also has some drawbacks, as MicroPPO's key limitation is its computational cost. Computing a projection during every forward pass is considerably more expensive than running a basic feed-forward neural network. Nonetheless, MicroPPO achieves the fastest convergence among all the DRL-based baselines as we guide the learning process by integrating the projection loss into the PPO loss objective.

Although sequential optimization performs well in our basic setting, it lacks flexibility as the complexity of the micro-grid increases. For instance, expanding the micro-grid model by adding a second ESS would necessitate revising the MILP formulation and introducing additional constraints. In contrast, MicroPPO is capable of handling large micro-grids more effectively. This is due to the branched structure of the actor network, where action choices in different dimensions are distributed across separate branches, allowing for some degree of independence in selecting actions. Thus, the number of network outputs grows only linearly with the number of action dimensions.

To illustrate, let us assume we want to add a second battery, B_2, to our micro-grid. In this case, we would need to make the following straightforward adjustments to the presented MDP and the actor network: (1) Add two dimensions to the action space, defined analogously to the second and third dimensions of a_t in Section 4.1, namely $a_t^{SC_3}$ and $a_t^{B_2G}$. (2) Add corresponding network branches to the actor network. (3) Add a second output to the meta-decision variable branch. (4) Add corresponding constraints to the set of constraints that define the space of feasible actions.

Hence, we plan to further investigate our model's capability of learning near-optimal policies for diverse objectives and increasingly complex micro-grids. A fruitful area for future work may involve integrating electric vehicles (EVs) that support vehicle-to-grid technology into our model, as they pose unique challenges for micro-grid operators. Most prominently, the intermittent unavailability of such an EV introduces a new dimension of complexity. A substantial question is how to deal with the fact that the EV's state of charge changes during unavailability due to driving or external charging. In this context, we will also focus on scenarios with limited operator control. In such partial control scenarios, challenges may mainly arise in achieving optimal coordination across all micro-grid components.

7 Conclusion

In this work, we propose MicroPPO, a Proximal Policy Optimization (PPO) based energy management system for minimizing the operational costs of a decentralized micro-grid. Unlike many existing deep reinforcement learning-based approaches, MicroPPO leverages the careful definition of a continuous, multi-dimensional action space for more nuanced control of power flows. Our approach introduces a novel actor network architecture comprising a shared module followed by several network branches: one for each action dimension and an additional branch for outputting meta-decision variables. Furthermore, our actor network integrates a differentiable projection layer, enabling the enforcement of system constraints during both the training and online execution.

Our experiments using real-world data show MicroPPO's convergence towards near-optimal policies obtained by solving a MILP formulation with perfect forecast information. Notably, our method outperforms other DRL algorithms employing discrete action spaces or implementing soft constraints. MicroPPO perfectly satisfies the system constraints over the entire horizon while learning to minimize operating costs within the safety set.

While our work highlights an approach to modeling sequential decision-making problems in energy systems using a continuous, multi-dimensional action space and guaranteeing agents' adherence to system constraints, we believe that this is only the beginning of a broader discussion on the close handling of multi-dimensional action spaces and the integration of domain knowledge into RL-based methods. In particular, allowing nuanced control in the presence of hard constraints will be critical to the real-world success of these methods in the context of energy systems.

Acknowledgments and Disclosure of Funding

This work was supported by the Ministry of Science, Research and the Arts Baden Württemberg, project Algorithm Engineering for the Scalability Challenge (AESC), by the German Research Foundation (DFG) as part of the Research Training Group GRK 2153: Energy Status Data – Informatics Methods for its Collection, Analysis and Exploitation, and by the Baden-Württemberg Foundation via the Elite Program for Postdoctoral Researchers. Additionally, this work was supported by the Helmholtz Association Initiative and Networking Fund on the HAICORE@KIT partition.

References

[1] Muhammad Fahad Zia, Elhoussin Elbouchikhi, and Mohamed Benbouzid. Microgrids energy management systems: A critical review on methods, solutions, and prospects. *Applied Energy*, 222:1033–1055, 2018.

[2] REN21. Renewables 2023 global status report collection., 2023.

[3] Hossein Shayeghi, Elnaz Shahryari, Mohammad Moradzadeh, and Pierluigi Siano. A survey on microgrid energy management considering flexible energy sources. *Energies*, 2019.

[4] Marvin Sigalo, Ajit Pillai, Saptarshi Das, and Mohammad Abusara. An energy management system for the control of battery storage in a grid-connected microgrid using mixed integer linear programming. *Energies*, 14, 2021.

[5] Luu Ngoc An and Tran Quoc-Tuan. Optimal energy management for grid connected microgrid by using dynamic programming method. In *2015 IEEE Power Energy Society General Meeting*, 2015.

[6] E.M. Craparo, Mumtaz Karatas, and Dashi Singham. A robust optimization approach to hybrid microgrid operation using ensemble weather forecasts. *Applied Energy*, 201, 2017.

[7] Ying Ji, Jianhui Wang, Jiacan Xu, Xiaoke Fang, and Huaguang Zhang. Real-time energy management of a microgrid using deep reinforcement learning. *Energies*, 12(12), 2019.

[8] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

[9] Chongyang Tian, Yigeng Huangfu, Rui Ma, Sheng Quan, Peng Li, Yonghui Zhang, and Jiang Wei. An improved rule-based energy management strategy for hybrid energy system with hydrogen cycle. In *2022 IEEE Industry Applications Society Annual Meeting (IAS)*, 2022.

[10] A. Kafetzis, Chrysovalantou Ziogou, K.D. Panopoulos, Simira Papadopoulou, Panos Seferlis, and Spyros Voutetakis. Energy management strategies based on hybrid automata for islanded microgrids with renewable sources, batteries and hydrogen. *Renewable and Sustainable Energy Reviews*, 134, 2020.

[11] Caisheng Wang and M. Hashem Nehrir. Power management of a stand-alone wind/photovoltaic/fuel cell energy system. *IEEE Transactions on Energy Conversion*, 23(3), 2008.

[12] Shivashankar Sukumar, Hazlie Mokhlis, Saad Mekhilef, Kanendra Naidu, and Mazaher Karimi. Mix-mode energy management strategy and battery sizing for economic operation of grid-tied microgrid. *Energy*, 118, 2017.

[13] Daniel Tenfen and Erlon Cristian Finardi. A mixed integer linear programming model for the energy management problem of microgrids. *Electric Power Systems Research*, 122, 2015.

[14] Yann Riffonneau, Seddik Bacha, Franck Barruel, and Stephane Ploix. Optimal power flow management for grid connected pv systems with batteries. *IEEE Transactions on Sustainable Energy*, 2(3), 2011.

[15] Erick O. Arwa and Komla A. Folly. Reinforcement learning techniques for optimal power control in grid-connected microgrids: A comprehensive review. *IEEE Access*, 8, 2020.

[16] R.A. Gupta and Nand Kishor Gupta. A robust optimization based approach for microgrid operation in deregulated environment. *Energy Conversion and Management*, 93:121–131, 2015.

[17] S. Surender Reddy, Vuddanti Sandeep, and Chan-Mook Jung. Review of stochastic optimization methods for smart grid. *Frontiers in Energy*, 11, 2017.

[18] Mario Petrollese, Luis Valverde, Daniele Cocco, Giorgio Cau, and José Guerra. Real-time integration of optimal generation scheduling with mpc for the energy management of a renewable hydrogen-based microgrid. *Applied Energy*, 166:96–106, 2016.

[19] Zhongwen Li, Chuanzhi Zang, Peng Zeng, and Haibin Yu. Combined two-stage stochastic programming and receding horizon control strategy for microgrid energy management considering uncertainty. *Energies*, 9(7), 2016.

[20] David Domínguez-Barbero, Javier García-González, Miguel A. Sanz-Bobi, and Eugenio F. Sánchez-Úbeda. Optimising a microgrid system by deep reinforcement learning techniques. *Energies*, 13(11), 2020.

[21] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, second edition, 2018.

[22] Grace Muriithi and Sunetra Chowdhury. Optimal energy management of a grid-tied solar pv-battery microgrid: A reinforcement learning approach. *Energies*, 14(9), 2021.

[23] Sunyong Kim and Hyuk Lim. Reinforcement learning based energy management algorithm for smart energy buildings. *Energies*, 11:2010, 2018.

[24] Yuchen Zhou, Sarah Henni, and Philipp Staudt. Managing intermittent renewable generation with battery storage using a deep reinforcement learning strategy. In *Wirtschaftsinformatik 2022 Proceedings*, 2022.

[25] Vincent François-Lavet, David Taralla, Damien Ernst, and Raphaël Fonteneau. Deep reinforcement learning solutions for energy microgrids management. In *European Workshop on Reinforcement Learning (EWRL 2016)*, 2016.

[26] Van-Hai Bui, Akhtar Hussain, and Hak-Man Kim. Double deep q-learning-based distributed operation of battery energy storage system considering uncertainties. *IEEE Transactions on Smart Grid*, 11(1), 2020.

[27] Tao Chen and Wencong Su. Local energy trading behavior modeling with deep reinforcement learning. *IEEE Access*, 6, 2018.

[28] Taha Abdelhalim Nakabi and Pekka Toivanen. Deep reinforcement learning for energy management in a microgrid with flexible demand. *Sustainable Energy, Grids and Networks*, 25, 2021.

[29] Hanna Krasowski, Jakob Thumm, Marlon Müller, Lukas Schäfer, Xiao Wang, and Matthias Althoff. Provably safe reinforcement learning: Conceptual analysis, survey, and benchmarking. *Transactions on Machine Learning Research*, 2023.

[30] Hou Shengren, Pedro P. Vergara, Edgar Mauricio Salazar Duque, and Peter Palensky. Optimal energy system scheduling using a constraint-aware reinforcement learning algorithm. *International Journal of Electrical Power Energy Systems*, 152, 2023.

[31] Bingqing Chen, Priya L. Donti, Kyri Baker, J. Zico Kolter, and Mario Bergés. Enforcing policy feasibility constraints through differentiable projection for energy optimization. In *Proceedings of the Twelfth ACM International Conference on Future Energy Systems*, e-Energy '21, page 199–210, 2021.

[32] Verena Jülch. Comparison of electricity storage options using levelized cost of storage (LCOS) method. *Applied Energy*, 183, 2016.

[33] Arash Tavakoli, Fabio Pardo, and Petar Kormushev. Action branching architectures for deep reinforcement learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18, pages 4131–4138. AAAI Press, 2018.

[34] Akshay Agrawal, Brandon Amos, Shane Barratt, Stephen Boyd, Steven Diamond, and J Zico Kolter. Differentiable convex optimization layers. *Advances in neural information processing systems*, 32, 2019.

[35] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. OpenAI Gym, 2016.

[36] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-Baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021.

[37] Adrian Beyertt, Paul Verwiebe, Stephan Seim, Filip Milojkovic, and Joachim Müller-Kirchenbauer. Felduntersuchung zu Behavioral Energy Efficiency Potentialen von privaten Haushalten. May 2020.

[38] Bundesnetzagentur für Elektrizität, Gas, Telekommunikation, Post und Eisenbahnen. Smard | marktdaten, 2023.

[39] Stefan Pfenninger and Iain Staffell. Long-term patterns of european pv output using 30 years of validated hourly reanalysis and satellite data. *Energy*, 114:1251–1265, 2016.

[40] Michele M. Rienecker, Max J. Suarez, Ronald Gelaro, Ricardo Todling, Julio Bacmeister, Emily Liu et al. Michael G. Bosilovich, Siegfried D. Schubert, Lawrence Takacs, Gi-Kong Kim, Stephen Bloom, Junye Chen, Douglas Collins, Austin Conaty, Arlindo da Silva, Wei Gu, Joanna Joiner, Randal D. Koster, Robert Lucchesi, Andrea Molod, Tommy Owens, Steven Pawson, Philip Pegion, Christopher R. Redder, Rolf Reichle, Franklin R. Robertson, Albert G. Ruddick, Meta Sienkiewicz, and Jack Woollen. Merra: Nasa's modern-era retrospective analysis for research and applications. *Journal of Climate*, 24(14):3624–3648, 2011.

A Appendix

A.1 Pseudocode for the Rule-based Baselines

This section contains the pseudocode for the two rule-based baselines, B3 (rule-based operation) and B4 (self-consumption pattern), against which we compare our approach. We provide the implementations of B3 and B4 in the GitHub repository.

Rule-based Operation (B3). Algorithm 1 presents the application of a pre-defined set of rules for a single time step. Baseline B3 prioritizes self-consumption and efficient battery usage based to energy prices. Surplus power generated by the PV system (i.e., Δg_t) is sold if current prices exceed the median of historical ones (cf. Line 6-7); otherwise, it is used to charge the battery, with any excess power being sold (cf. Line 9). Power deficits are addressed by storage if available, supplemented by purchased energy. Thereby, B3 uses the battery-related `Charge` and `Discharge` routines outlined in Algorithm 2.

Algorithm 1: Baseline B3 (Rule-based Operation)

```
Input: $l_t^H$, $g_t^{PV}$, $c_t$, $SOC_t^B$, $SOC_{min}^B$, $\eta_{ch/dis}^B$, $\Psi = [c_{t-1}, \dots, c_0]$
Output: $p_{in,t}^B$, $p_{out,t}^B$, $p_{in,t}^G$, $p_{out,t}^G$
1  $\Delta l_t \leftarrow -l_t^H$
2  $p_{in,t}^B, p_{out,t}^B, p_{in,t}^G, p_{out,t}^G \leftarrow 0$
3  if $g_t^{PV} \geq l_t^H$ then
4      $\Delta l_t \leftarrow 0$
5      $\Delta g_t \leftarrow g_t^{PV} - l_t^H$
6      if $c_t > median(\Psi)$ then
7          $p_{out,t}^G \leftarrow p_{out,t}^G + \Delta g_t$
8      else
9          $p_{in,t}^B, p_{out,t}^G \leftarrow$ Charge($\Delta g_t$, $p_{in,t}^B$, $p_{out,t}^G$, $\eta_{ch}^B$)
10 else
11     $p_{dis,t}^B \leftarrow (SoC_{t-1}^B - SoC_{min}^B) \cdot \frac{U^B}{\Delta t}$
12     if $p_{dis,t}^B > 0$ then
13         $p_{out,t}^B \leftarrow$ Discharge($p_{dis,t}^B$, $l_t^H$, $p_{out,t}^B$, $\eta_{dis}^B$)
14         $\Delta l_t \leftarrow \Delta l_t + p_{out,t}^B \cdot \eta_{dis}^B$
15     $\Delta l_t \leftarrow \Delta l_t + g_t^{PV}$
16 $p_{in,t}^G \leftarrow p_{in,t}^B + \max(0, -\Delta l_t)$
17 $p_{out,t}^G \leftarrow p_{out,t}^B + \max(0, \Delta l_t)$
18 $\Psi \leftarrow \Psi \cup [c_t]$
19 return $p_{in,t}^B, p_{out,t}^B, p_{in,t}^G, p_{out,t}^G$
```

Algorithm 2: Battery-related `Charge` and `Discharge` routines

```
Function Charge($\Delta g_t$, $p^B_{in,t}$, $p^G_{out,t}$, $\eta^B_{ch}$):
    $p^{PV}_{B,t} \leftarrow$ B.charge$(\Delta g_t) \cdot \frac{1}{\eta^B_{ch}}$
    $\Delta g_t \leftarrow \Delta g_t - p^{PV}_{B,t}$
    $p^B_{in,t} \leftarrow p^B_{in,t} + p^{PV}_{B,t}$
    if $\Delta g_t > 0$ then
        $p^G_{out,t} \leftarrow p^G_{out,t} + \Delta g_t$
    return $p^B_{in,t}, p^G_{out,t}$

Function Discharge($p^B_{dis,t}$, $l^H_t$, $p^B_{out,t}$, $\eta^B_{dis}$):
    $w_t \leftarrow l^H_t \cdot \frac{1}{p^B_{dis,t}}$
    $p^B_{H,t} \leftarrow$ B.discharge$(w_t)$
    $p^B_{out,t} \leftarrow p^B_{out,t} + p^B_{H,t}$
    return $p^B_{out,t}$
```

Self-consumption Pattern (B4). Algorithm 3 illustrates a variant of baseline B3 that is commonly used in practice. Like B3, baseline B4 prioritizes the self-consumption of generated or stored power and executes the charging and discharging routines from B3 (cf. Line 6 and 10). However, unlike B3, it does not consider current energy prices.

Algorithm 3: Baseline B4 (Self-consumption pattern)

```
Input: $l^H_t$, $g^{PV}_t$, $c_t$, $SOC^B_t$, $SOC^B_{min}$, $\eta^B_{ch/dis}$
Output: $p^B_{in,t}$, $p^B_{out,t}$, $p^G_{in,t}$, $p^G_{out,t}$

 1 $\Delta l_t \leftarrow -l^H_t$
 2 $p^B_{in,t}, p^B_{out,t}, p^G_{in,t}, p^G_{out,t} \leftarrow 0$
 3 if $g^{PV}_t \geq l^H_t$ then
 4     $\Delta l_t \leftarrow 0$
 5     $\Delta g_t \leftarrow g^{PV}_t - l^H_t$
 6     $p^B_{in,t}, p^G_{out,t} \leftarrow$ Charge($\Delta g_t$, $p^B_{in,t}$, $p^G_{out,t}$, $\eta^B_{ch}$)
 7 else
 8     $p^B_{dis,t} \leftarrow (SoC^B_{t-1} - SoC^B_{min}) \cdot \frac{U^B}{\Delta t}$
 9     if $p^B_{dis,t} > 0$ then
10         $p^B_{out,t} \leftarrow$ Discharge($p^B_{dis,t}$, $l^H_t$, $p^B_{out,t}$, $\eta^B_{dis}$)
11         $\Delta l_t \leftarrow \Delta l_t + p^B_{out,t} \cdot \eta^B_{dis}$
12     $\Delta l_t \leftarrow \Delta l_t + g^{PV}_t$
13 $p^G_{in,t} \leftarrow p^G_{in,t} + \max(0, -\Delta l_t)$
14 $p^G_{out,t} \leftarrow p^G_{out,t} + \max(0, \Delta l_t)$
15 return $p^B_{in,t}, p^B_{out,t}, p^G_{in,t}, p^G_{out,t}$
```

A.2 Hyperparameters of the DRL-based Baselines

In this section, we detail the hyperparameter configurations of the DRL-based baseline B5-B7 used in our experiments.

Similar to MicroPPO, we train B5-B7 for 1,209,600 steps, which corresponds to 7,200 episodes, each consisting of 168 steps. We provide an implementation of these models using the `OpenAI Gym` framework[35] and the `Stable Baselines 3` library [36] in the repository on GitHub.

Table 2 and Table 3 present the hyperparameters of the PPO-based baselines (B5 and B7) and the DQN-based baseline B6, respectively. For B6 and B7, following [24], we discretize the action space in each dimension into tenths. For example, if a continuous action dimension a^I ranges from 0 to 1, the corresponding discretized sub-action space is $a^I_{disc} = \{0, 0.1, \dots, 0.9, 1\}$, comprising eleven sub-actions.

Table 2: Parameters of the PPO-based baselines (B5, B7)

Hyperparameter	**B5 (PPO-C)**	**B7 (PPO-D)**
λ_v	0.5	
ϵ	0.2	
γ	0.99	
Batch size	168 steps	
Update frequency	every 7 steps	
Optimizer	Adam	
Learning rate	5×10^{-4}	
Exploration rate	0.1	
Actor network	Multi-layer perceptron (2 layers, 64 units each, ReLU activation)	
Critic network	Multi-layer perceptron (2 layers, 32 units each, ReLU activation)	
λ_{PV}	10	-
λ_B	5	-

Table 3: Parameters of the DQN-based baseline (B6)

Hyperparameter	**B6 (DQN)**
γ	0.99
Replay buffer size	1.0×10^6
Batch size	168 steps
Learning start	after 24 steps
Update frequency	every 4 steps
Update frequency (target)	every 168 steps
Optimizer	Adam
Learning rate	5×10^{-4}
Networks	Multi-layer perceptron (2 layers, 64 units each, ReLU activation)