EnEnv 1.0: Energy Grid Environment for Multi-Agent Reinforcement Learning Benchmarking

Dominik Jacek Bogucki
Institute of Fundamental Technological Research, Polish Academy of Sciences
IDEAS NCBR
Warsaw, Poland
dominik.bogucki@ideas-ncbr.pl

Łukasz Lepak
Institute of Computer Science, Warsaw University of Technology
IDEAS NCBR
Warsaw, Poland
lukasz.lepak.dokt@pw.edu.pl

Sonam Parashar
IDEAS NCBR
Warsaw, Poland
sonam.parashar@ideas-ncbr.pl

Bartłomiej Błachowski
Institute of Fundamental Technological Research, Polish Academy of Sciences
Warsaw, Poland
bblach@ippt.pan.pl

Paweł Wawrzyński
IDEAS NCBR
Warsaw, Poland
pawel.wawrzynski@ideas-ncbr.pl

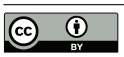
ABSTRACT

Multi-agent reinforcement learning (MARL) offers prospects of efficient control in large distributed systems such as complex energy grids. The development of MARL algorithms is hampered by a scarcity of realistic benchmarks. In this paper, we introduce EnEnv 1.0 — a simulation benchmark for MARL in modern energy grids. EnEnv 1.0 is a set of environments in which the energy grids are simulated with uncontrollable renewable energy sources, fossil fuel generators, and consumers. The role of learning agents is to control and coordinate batteries in a distributed Battery Energy Storage System (BESS) based on readouts such as weather forecasts and load demand forecasts. The energy grids in EnEnv 1.0 are based on standard test systems of different topological structures. These include the modified standard IEEE 33, Illinois 200, and PEGASE 89 bus systems. These networks are adjusted to serve as the MARL benchmark by introducing real weather observations, demand data for European locations, and software interfaces that enable coupling with a number of existing implementations of MARL algorithms, as well as single-agent reinforcement learning (SARL) algorithms. In the experimental study, we verify the performance of a catalog of MARL and SARL methods on EnEnv 1.0.

KEYWORDS

Multi-Agent Reinforcement Learning; Energy Grid; Battery Energy Storage System

ACM Reference Format:
Dominik Jacek Bogucki, Łukasz Lepak, Sonam Parashar, Bartłomiej Błachowski, and Paweł Wawrzyński. 2025. EnEnv 1.0: Energy Grid Environment for Multi-Agent Reinforcement Learning Benchmarking. In *Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Detroit, Michigan, USA, May 19 – 23, 2025*, IFAAMAS, 10 pages.



Proc. of the 24th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2025), Y. Vorobeychik, S. Das, A. Nowé (eds.), May 19 – 23, 2025, Detroit, Michigan, USA. © 2025 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org).

1 INTRODUCTION

Multi-agent reinforcement learning (MARL) [8, 38–40, 60] promises efficient control in large-scale systems through the collective learning of many agents in dynamic environments. Multiple agents interact within a shared environment to learn optimal policies. MARL is defined as the process where agents optimize their behaviors based on the history of their interactions with the environment, i.e., on their experience. Research in this area aims to design methods that enable agents to learn quickly, i.e., from minimal experience. The challenges and complexities in using MARL for controlling multi-agent systems include coordination among agents, non-stationary environments due to evolving agent policies, and scalability issues. Advanced MARL approaches often involve sophisticated techniques such as decentralized training, communication protocols among agents, and hierarchical methods to manage multi-agent interactions as effectively as possible. Theoretical advancements and practical applications of MARL demonstrate its potential in diverse fields ranging from robotics [41] and autonomous driving [1, 49, 61] to telecommunications [13, 28].

Development in MARL and other areas of artificial intelligence is based on the availability of benchmark environments in which various learning methods can be verified and compared. These benchmarks should ideally represent challenging real-life problems whose solutions could be profitably implemented. However, few benchmarks in MARL represent real-life problems to which MARL could actually be applied.

Energy Grids (EGs) present complex systems that naturally define a number of distributed control problems. These problems have become especially interesting recently due to the rapidly growing presence of renewable energy sources (RES). The output of these sources is weather-dependent, not controllable, and only predictable to a certain extent, which causes numerous operational challenges,

such as managing active and reactive power to maintain voltage and frequency in real-time within permissible limits [6, 14, 34] and ensuring economic performance [29, 35, 36]. Another challenge is controlling the back-feeding or reverse power flow to external networks in case of surplus renewable energy generation [17, 24, 66].

Replacing controllable power generators based on fossil fuels with uncontrollable RES raises an issue of potential mismatch between power supply and demand within the modern grids, causing instability in terms of voltage and frequency and potential outages. There are two general solutions to this issue. The first one is to adjust the demand to match the supply, that is, to consume energy when it is being produced. The second one is based on a Battery Energy Storage System (BESS): The storing of energy when it is produced and releasing it when there is demand for it. In this paper, we focus on the second solution and consider MARL for efficient control of component batteries of a distributed BESS. It has been noticed that there is no such environment available to simulate modern energy grids using MARL, which includes built-in renewable energy sources and batteries integrated into large energy grids. Therefore, we propose an environment with a MARL interface to simulate modern energy grids.

The contributions of this paper are as follows:

(1) We formalize the problem of distributed BESS control as a Multi-Agent Markov Decision Problem.
(2) We introduce EnEnv 1.0, a benchmark for control learning algorithms in which the role of agents is to control distributed BESS. The nature of the algorithms may be diverse, but it is especially suitable for MARL and SARL.
(3) We report the application of a catalog of MARL and SARL algorithms in EnEnv 1.0.

2 RELATED WORK

Multi-Agent Reinforcement Learning (MARL)

Single-Agent Reinforcement Learning (SARL) [54] addresses trial-and-error learning of sequential decision-making under uncertainty. In Multi-Agent Reinforcement Learning (MARL) [39], a group of agents learns to act in the same dynamic environment, collectively modifying its state. There are a number of different settings for MARL, depending on whether the agents operate synchronously or asynchronously, have common goals or adverse ones, and other circumstances. In this paper, we focus on the synchronous and cooperative MARL, with a single reward value for all agents at every time instance.

MARL has been addressed by extending well-established SARL algorithms: Multi-Agent SAC [58], Multi-Agent PPO [63], Multi-Agent DDPG [33].

Generally, cooperative MARL adopts a centralized training with a decentralized execution (CTDE) paradigm, which suffers from the global action–value function, whose complexity grows exponentially with the number of agents [21]. Therefore, action–value function decomposition is a fundamental problem in MARL. Existing decomposition methods include VDN [53], QMIX [46], QTRAN [51], Weighted QMIX [45], QPLEX [56], Qatten [62] and NA^{2}Q [32].

Another fundamental problem in the cooperative MARL is coordinating agents' activity with a certain communication protocol. A number of solutions have been proposed for that purpose, including CommNet [52], TarMAC [5], NDQ [57], GA-Comm [30], IS [20], MAIC [64], DHCG [31] and [15], also solutions in which the agents communicate in natural language: Symbolic PPO [59], TWOSOME [55], and Verco [27].

MARL benchmarks

Two MARL frameworks support the most recent computational technologies: MARLlib [19] and BenchMARL [3]. They both support running a catalog of MARL algorithms in most benchmark environments ever used in MARL research. MARLlib is provided with the SMAC, MPE, GRF, MAMuJoCO, and MAgent environments, while BenchMARL is provided with VMAS, SMACv2, MPE, Pursuit, Waterworld, and MeltingPot. Each environment comes with several different tasks.

StartCraft Multi-Agent Challenge (SMAC), [48] and SMACv2 [10] are based on SartCraft, a multiplayer strategic game originally introduced for human players in 1998. Multi Particle Environment (MPE) [33] is a communication oriented environment where particle agents can (sometimes) move, communicate, see each other, push each other around, and interact with fixed landmarks. Google Research Football (GRF) [25] is an environment where agents are trained to play football in an advanced, physics-based 3D simulator. Multi-Agent Mujoco (MAMuJoCo) [44] is an environment for continuous cooperative multi-agent robotic control. Based on the popular single-agent robotic MuJoCo control suite, it provides a wide variety of novel scenarios in which multiple agents within a single robot have to solve a task cooperatively. MAgent [65] is an environment where large numbers of pixel agents in a grid world interact in battles or other competitive scenarios. Vectorized Multi-Agent Simulator (VMAS) [2] is a simulator comprised of a vectorized 2D physics engine and a set of multi-robot scenarios. In Multiwalker [16], three planar robots collectively carry a package on them. Pursuit [16], aka Predator-Pray [33], is a grid world in which slowly moving pursuers are rewarded for surrounding faster-moving evaders. Waterworld [16] is a square in which agents pursue food targets and avoid poison targets, both moving. MeltingPot [26] has 18 variants; it is a grid world in which the agents observe only their immediate surroundings and perform one of 6 actions.

Papoudakis et al. [42] analyze the performance of various MARL algorithms on five benchmarks: MPE and SMAC mentioned above, also grid-world based Level-Based Foraging and Multi-Robot Warehouse, and Repeated Matrix Games with 2 players, 3 actions and predefined payoff matrices.

The Overcooked [27] is a 7x7 grid-size kitchen where two agents communicate in natural language to make different salads with the provided raw materials and tools.

To summarize this section, we may conclude that the existing MARL benchmark may be challenging problems, but they are relatively far from real-life applications. This contrasts SARL benchmarks that represent difficult real-life problems, especially in robotics. Therefore, the environment proposed in this paper is intended to be a benchmark similar to the actual control problem in power systems.

SARL and MARL for energy storage control

Various control problems in power systems have already been addressed with SARL and MARL. Samende et al. [47] applied the MADDPG algorithm to optimize the scheduling of the hybrid energy storage system and energy demand in real-time for grid-connected microgrid. The model-free reinforcement learning algorithms that completely ignore the physics-based modeling of the energy grid compromise scalability challenges. Krishnamoorthy et al. [22] address this issue and proposes imitation learning-based improvements in deep SARL to provide a good initial policy that increases training efficiency to solve the single agent battery storage dispatch problem for frequency regulation in the power distribution systems. Pei et al. [43] address the voltage regulation and peak demand in real feeders through two agents called PV inverter and battery storage; for this, they proposed two-stage deep reinforcement learning. A single-agent energy management problem for BESS and on-grid supply is addressed using the DQN algorithm in [50]. Krishnamoorthy et al. [23] propose an OpenDSS-RL wrapper for voltage regulation power distribution grid considering step voltage regulators, capacitor banks, and batteries as agents.

Fan et al. [12] proposes PowerGym, an open-source SARL environment for Volt-Var and reactive power control in Power distribution systems combining OpenDSS and OpenAI gym considering transformer taps, batteries, capacitors, and regulators as agents. On the other hand, Gym-ANM [18] is a simulator that allows users to design a power grid with lines, generators, appliances, and storage and simulate their work with various control mechanisms, including SARL.

It is observed that most of the work presented in the literature considers voltage and reactive power control in power distribution systems using SARL algorithms. However, the dynamics of the power grid with distributed BESS are more profound because of the changing state of charge of the batteries. Also, control of this system is naturally distributed because of the locality of available information. Therefore, MARL for distributed BESS control in the distribution networks for energy management considering power losses and reverse power flow is not explored sufficiently and needs to be addressed carefully. In this order, we propose a MARL-based framework for cooperative distributed BESS control in the power distribution system to minimize transmission loss while ensuring maximum renewable energy utilization.

3 MULTI-AGENT REINFORCEMENT LEARNING

In this paper, we discuss the problem of MARL using the formalism of a cooperative Multi-Agent Partially Observable Markov Decision Process (MAPOMDP). A MAPOMDP is defined by a tuple, $\langle \mathbb{N}, \mathbb{S}, \mathbb{A}, \mathcal{P}, O, \mathcal{R} \rangle$, where $\mathbb{N} = \{1, \ldots, n\}$ is a team of agents, $\mathbb{S}$ is a space of environmental states, $\mathbb{A} = \mathbb{A}^1 \times \cdots \times \mathbb{A}^n$ is a space of team actions of the agents, $\mathcal{P}$ is the state transition probability, O is the observation function, and $\mathcal{R}$ is the reward function. We generally assume that all the considered spaces are continuous, and in particular, $\dim(\mathbb{A}^i) = d$ for all i. At the time $t = 1, 2, \ldots$ each i-th agent makes an observation, $o_t^i = O^i(s_t)$, of the environment state s_t and based on this observation, performs an action, $a_t^i \in \mathbb{A}^i$.

The team action $a_t = [(a_t^1), \ldots, (a_t^n)]$ impacts the next environment state $s_{t+1} \sim \mathcal{P}(\cdot|s_t, a_t)$ and the reward $r_t = \mathcal{R}(s_{t+1})$ is given collectively to all agents.

The agents choose their actions based on their policies, $a_t^i \sim \pi^i(\cdot|o_t^i)$, where π^i is the policy of i-th agent. The goal of the agents' learning is to optimize their team policy $\pi = [\pi^1, \ldots, \pi^n]$ for the team to expect in each environment state s the highest sum of discounted rewards

$$\mathcal{V}^{\pi}(s) = E\left(\sum_{i \geq 0} \gamma^i r_{t+i} \middle| s_t = s, \text{policy in use} = \pi\right),$$

where $\gamma \in [0, 1]$ is the discount factor.

4 CONTROL OF DISTRIBUTED BESS AS A MULTI-AGENT MARKOV DECISION PROCESS

In this section, we characterize the Energy Grid (EG) and how it defines a MAPOMDP problem in the previous section. EG can be represented by a graph. Its edges represent transmission lines. In nodes of the graph, there are power consumers and power sources, i.e., power generators based on fossil fuels (FG), wind turbines (WT), photovoltaic farms (PV), as well as batteries (Battery Energy Storage, BES). Together, the batteries create a distributed Battery Energy Storage System (BESS). EG is connected to the external world through a node called *slack bus*. EG may purchase energy from the external world and sell energy to the external world. However, because the buy price is higher than the sell price, it is the most efficient for EG to maximize its self-sufficiency. We adopt the following assumptions about the operation of EG:

(1) The power load demand (consumption) changes according to the statistical profile of households.
(2) Power generated by the FGs is just constant.
(3) Power generated by the WTs and PVs depends on the weather.
(4) The only controllable devices in EG are BESs. The policy of their control is optimized with RL.
(5) The energy exchange between EG and the slack bus is a residual of energy generated and absorbed inside EG.
(6) The energy generation and demand inside EG are calibrated so that their yearly sums are approximately equal. The role of BESS control can be understood as minimizing the momentary energy imbalances.

The volatility of WT and PV generation results in scenarios where power generation in the system is surplus or deficit, causing an imbalance in generation and load demand. However, the system operator needs to manage the operation of the EG to achieve a balance between the volatile generation from the WT and PV and fluctuating load demand at a particular time period under safe operating conditions, such as voltage and thermal limits at each bus and line given by (1,2) are not violated.

$$V^{min} \leq V_{j,t} \leq V^{max} \tag{1}$$

$$I_{k,t} \leq I_k^{max} \tag{2}$$

where, V^{min} and V^{max} are the upper and lower limit of bus voltage. $V_{j,t}$ is the actual voltage of j-th bus at t-th hour. $I_{k,t}$ is the current

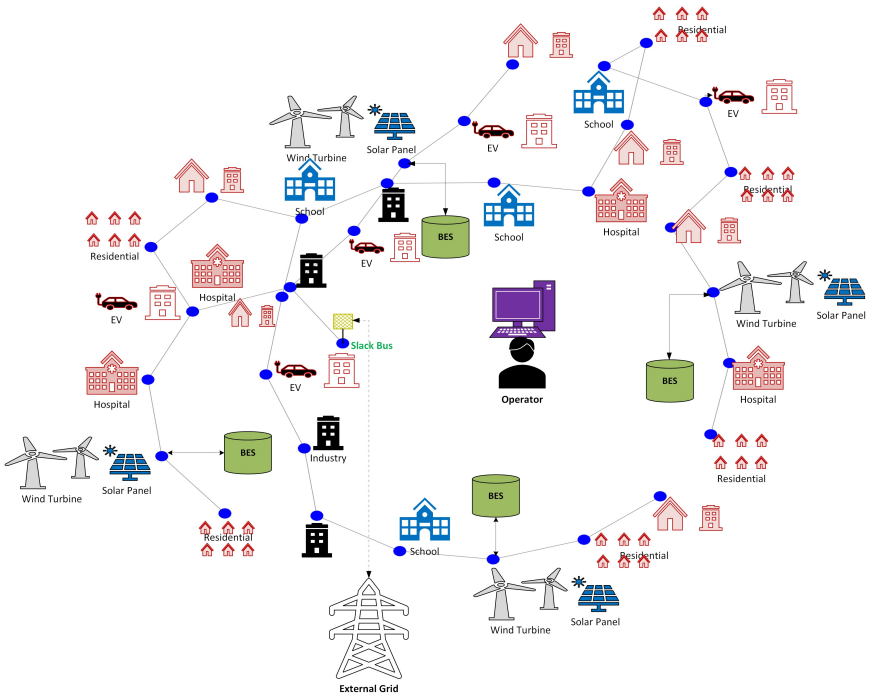


Figure 1: Distributed BES control interactive network setting

flows in k-th line at t-th hour. I_k^{max} is the maximum limit for the current beyond which the line melts.

The operation and control of BES at time step t relies on the State of Charge (SoC) available at the current time step and various SoC-related constraints. The future SoC at the $(t+1)$ time step is given by (3):

$$C_{i,(t+1)} = \eta_s \cdot C_{i,t} + \Delta t \cdot P_{i,t}^{ch} \cdot \eta_c - \frac{\Delta t \cdot P_{i,t}^{dis}}{\eta_d} \tag{3}$$

such that $P_{i,t}^{ch} \cdot P_{i,t}^{dis} = 0$, where, $P_{i,t}^{ch}$ and $P_{i,t}^{dis}$ are the charging and discharging power of i-th BES agent respectively. η_s, η_c and η_d are standing loss efficiency, charging and discharging efficiency, respectively, and $\Delta t = 1$ is the time step of charging or discharging of the i-th BES.

Also, the charging/discharging operation of BES should not violate the SoC constraint given by (4)

$$C_i^{min} \leq C_{i,(t+1)} \leq C_i^{max} \tag{4}$$

where, C_i^{min} and C_i^{max} are the minimum and maximum SoC limit of i-th BES agent respectively.

4.1 Problem formulation

We model the problem of control of distributed energy storage by a cooperative MAPOMDP, introduced in Section 3. In this MAPOMDP, time t defines hours. An agent controls a BES in the grid by applying an action, $a_t^i \in [-1, 1]$, which defines how much energy will be absorbed, within the following hour, from the grid to the BES or released in the opposite direction. i-th agent chooses its actions based on the regional information relevant to the operation of this agent and the aggregate information relevant to the operation of the whole EG.

At each hour t, the load demand in the grid nodes is set, as is the generation in energy sources. The agents determine how much energy is stored into or released from the BESS. Based on this input, the energy flow over the system is determined using the Newton-Raphson numeric procedure to satisfy the constraints (1) and (2) and calculate the total transmission loss within the grid. In the process, an amount of energy P_t^{slack} is determined to be purchased from (or sold to) the slack bus to balance the system.

4.1.1 Reward function. Since the grid typically sells energy at a lower price than it purchases, agents should avoid exchanging energy with the slack bus. Instead, they should charge the BESS during moments of energy surplus and discharge it during moments of energy deficit. Additionally, they should coordinate their operations effectively to minimize transmission losses and internal inefficiencies within the grid. Therefore, we formulate the reward function as the sum of the following components:

- Battery Power Loss, P^b: This is defined as the sum of the battery's internal power loss due to self-discharge and external power conversion losses. The battery power loss for i-th agent is given by (5)

$$\begin{aligned} P_{i,t}^b = & C_{i,t} \cdot (1 - \eta_s) + \Delta t \cdot P_{i,t}^{ch} \cdot (1 - \eta_c) \\ & + \Delta t \cdot P_{i,t}^{dis} \cdot \left(\eta_d^{-1} - 1\right) \end{aligned} \tag{5}$$

  which corresponds to how the battery state of charge is updated in (3).

The total battery power loss at t-th time is given by (6)

$$P_t^b = \sum_{i=1}^{B} P_{i,t}^b \tag{6}$$

where P_t^b is the total battery power loss at t-th time, a sum of losses of each battery, and B is the number of batteries in the system.

- Transmission Loss, P^L: These are the power losses that occur in the transmission lines during power flow in the energy grid. The transmission losses are obtained by executing the Newton-Raphson (NR) load flow algorithm.
- Slack Bus Loss, P^s: We define slack bus losses as the financial losses that occur due to energy exchange. For instance, the grid operator purchases a unit of energy for $p > 0$ and sells the same unit for $q \cdot p$, where $q \in (0, 1)$. In this case, whenever a unit of energy is purchased or sold, we assume it to incur a cost of $(1 - q)/2$ to the grid operator because if it is sold (or purchased), the same amount of energy will eventually need to be purchased (or sold), thus incurring a total cost proportional to $(1 - q)$. Therefore, the slack bus loss is given by (7)

$$P_t^s = |P_t^{slack}| \cdot \frac{(1-q)}{2} \tag{7}$$

where q is the cost of selling the energy relative to its buying, and P_t^{slack} is the amount of energy sold or purchased by the grid operator. If $P_t^{slack} > 0$, energy is sold; otherwise, it is purchased. In our experiments, we assume $q = 0.5$.

The mathematical expression for the reward function is given by

$$\mathcal{R} = -\left(\frac{P_t^b + P_t^L + P_t^s}{r^f}\right) \tag{8}$$

where r^f is a reward standardization factor described in Appendix C.

5 ENENV 1.0. PROPOSED BENCHMARK

EnEnv 1.0 code is available in the official GitHub repository [7].

5.1 Energy grid networks

To test the efficiency of the RL algorithms on complex energy grid networks, the proposed benchmark uses various standard EGs available in Pandapower, the IEEE 33, PEGASE 89, and Illinois 200 bus systems. The IEEE 33 is a low/medium voltage network and exhibits a radial structure consisting of loads on nodes. At the same time, the Illinois 200 and PEGASE 89 are high-voltage networks exhibiting a meshed structure with several fossil fuel-based generators, loads, and static generators located on different nodes. The default standard systems are not integrated with renewable energy sources and BESs. They are steady in nature, which is not suitable for studying realistic operating scenarios where the operator needs to control BES in coordination with uncertain and dynamic generation from RES, as well as load demand. For this purpose, we transform these networks into dynamic settings, as explained in Section 5.2. Also, we proposed suitable RL interfaces to simulate these modern grid environments.

5.2 The data

We fit the topology of the standard IEEE 33, Illinois 200, and PEGASE 89 systems to the entire map of Germany and obtained weather data for the grid bounded by extreme eastern, western, northern, and southern coordinates. Additionally, we considered a number of distributed BES agents to be in control in different grid networks, such as 4, 11, and 37 in IEEE 33, PEGASE 89, and Illinois 200, respectively, at various locations. These are discussed in detail further in this subsection.

5.2.1 Predictions. Observations available to the agents include predictions of future load, wind, and solar irradiation. In training and evaluation, we use the following method of randomized generation of predictions. This method addresses the following requirements:

(1) The predictions are similar to the predicted values.
(2) The relative error increases with the prediction horizon.
(3) The above relative error equals ε for a prediction horizon of H, where ε and H are based on the reported accuracy of the predictions of the considered values.
(4) Predictions of the same value available at adjacent times are similar.

Let $\widehat{x}_{t+h|t}$ be a prediction of the value x_{t+h} available at time t; h is the prediction horizon. We generate a sequence of independent prediction deviations, $\xi_t \sim N(0, 1)$. We assume the prediction

$$\widehat{x}_{t+h|t} = x_t \left(1 + \tanh\left(z_{t,h}\right)\right), \tag{9}$$

where

$$z_{t,h} = \sigma \sum_{i=1}^{h} \xi_{t+i}/(i+1), \quad \sigma = \varepsilon / \sqrt{\sum_{i=1}^{H} (i+1)^{-2}} \tag{10}$$

The above requirements 1-4 are satisfied as follows. As long as the variance of $z_{t,h}$ is small, $\widehat{x}_{t+h|t}$ is close to x_{t+h}, because tanh is continuous and $\tanh(0) = 0$. The variance of $z_{t,h}$ equals $\sigma^2 \sum_{i=1}^{h} (i+1)^{-2}$, which increases with the prediction horizon h. For $h = H$, it is ε^2. Thus, the relative prediction error for a horizon of H equals ε, as required. The predictions $\widehat{x}_{t+h|t}$ and $\widehat{x}_{t+h|t+1}$ are close to each other because their respective determinants $z_{t,h}$ and $z_{t+1,h-1}$ are sums of overlapping sets of random components.

5.2.2 Load demand. The hourly load demand of the network is obtained by considering the nominal loads of the system and the realistic historical hourly load demand curve for Germany, sourced from ENTSO-E, for the years 2017 to 2024 [11]. Furthermore, we obtain the hourly load data for each bus in the network by processing it in the following two stages:

(1) Load distribution: At this stage, the hourly load curve for all load buses is obtained by scaling the nominal loads by an auto-regressive load scaling factor $P_{j,t}^{d,scaling}$, calculated for each bus as follows.

$$X_{j,t} = X_{j,(t-1)} \cdot \alpha + \xi_{j,t} \cdot \sigma \cdot \sqrt{1 - \alpha^2} \tag{11}$$

where, $X_{j,t}$ and $\xi_{j,t} \sim N(0, 1)$ is an autoregressive noise and prediction deviations for j-th bus at t-th hour respectively , $\alpha = 0.9$ and $\sigma = 0.3$

$$Y_{j,t} = \frac{\kappa_t \left(P_j^{d,nom} \cdot \exp(X_{j,t}) \right)}{\sum_i P_i^{d,nom} \cdot \exp(X_{i,t})} \tag{12}$$

where

$$\kappa_t = \frac{P_t^d}{P^{peak}}, \tag{13}$$

$$P_{j,t}^{d,actual} = Y_{j,t} \cdot P_t^d, \tag{14}$$

where, $Y_{j,t}$ is the scaling factor for load demand. κ_t, P_t^d, and P^{peak} are the hourly load factor, hourly load demand, and peak demand, respectively, obtained from the Germany load curve. $P_j^{d,nom}$ and $P_{j,t}^{d,actual}$ are the nominal load demand and actual hourly load demand given for a j-th bus of the power network.

(2) Load Forecasting: After distributing the load demand to all the load buses, the load demand predictions $\widehat{Pd}_{j,t+h|t}$ are based on the actual demand $P_{j,t+h}^{d,actual}$ according to (9), with the average relative error $\varepsilon = 2.3\%$ for the prediction horizon $H = 24[hours]$.

5.2.3 Fossil fuel generators. Considering the convenience of power plant operators and the techno-economic operations and constraints of fossil fuel generators (FGs), we assumed that FGs are scheduled to supply a fixed base load for the network.

Also, considering the current scenario in Germany, where renewable energy penetration in the system is around 60%, and FGs are serving 40% of the load demand, we set the power generation of FGs as follows:

$$P^{g,FG} = \tau_f \cdot P^{d,avg} \tag{15}$$

where $\tau_f = 0.4$ and $P^{g,FG}$ is the total power generation from FGs. Furthermore, we calculate the contribution factor λ to allocate the scheduled generation to each FG in the network.

$$\lambda_i = \frac{P_i^{g,FG,nom}}{\sum_j P_j^{g,FG,nom}} \tag{16}$$

where $P_i^{g,FG,nom}$ and λ_i are the nominal power produced given in the basic network and contribution factor of the i-th FG, respectively.

Then,

$$P_i^{g,FG,actual} = \max(P_i^{g,FG,min}, P^{g,FG} \cdot \lambda_i) \tag{17}$$

where $P_i^{g,FG,actual}$ and $P_i^{g,FG,min}$ represent the actual generated power and the minimum power generation limit of the i-th FG, respectively.

5.2.4 Renewable generation. The uncertain wind speed and solar radiation cause fluctuations in the hourly power output of wind turbines and solar panels. Therefore, wind speed and solar irradiation are predicted as follows:

(1) Wind Forecasting: The wind forecasts $\widehat{v}_{j,t+h|t}$ are based on their actual values $v_{j,t+h}^{actual}$ according to (9) with $\varepsilon = 25\%$ for forecast horizon of $H = 72$ [9].

(2) Solar Forecasting: The solar irradiation forecasts $\widehat{G}_{j,t+h|t}$ are based on their actual values $G_{j,t+h}^{actual}$ according to (9) with $\varepsilon = 12\%$ for single hour forecast $H = 1$ [4]. The irradiation predictions are also limited by their maximum historical values at a given hour of the day and week of the year. This capping procedure allows us to avoid forecasting values greater than clear-sky values.

(3) Renewable generators placement: In the proposed benchmark, the PQ type of wind and solar distributed generators (DGs) are considered in the system. These generators can supply active (P) and reactive power (Q). However, these generating units can generate very low amount of reactive power. Generally, solar panels cannot produce reactive power by themselves; these generating units are inverter-based resources (IBR), meaning the inverter connected to these units can provide some reactive power control by adjusting its parameters. Similarly, the reactive power generated by wind turbines is also very low. Therefore, in this work, we consider the total reactive power support from these devices to be very low and set to 25% of total active power generation. The total WT and PV units in the systems are 4, 11, and 37 units within IEEE 33, PEGASE 89, and Illinois 200, respectively. In IEEE 33 bus systems, these units are placed on bus no. 5, 14, 24 and 30 [37] whereas, in PEGASE 89 and Illinois 200 bus system, units are placed together on randomly selected load buses within the network.

60% of the total demand is assumed to be served by renewable energy generators (RGs), i.e., wind turbines (WT) and solar panels (PV) with equal share. The peak power of all the WT and PV installations are determined, respectively, as follows:

$$P^{WT,peak} = \tau_w \cdot P^{d,avg} \tag{18}$$

$$P^{PV,peak} = \tau_s \cdot P^{d,avg}, \tag{19}$$

where $P^{d,avg}$ is the average load demand of the system, $\tau_w = 0.3 \cdot 2.45$ and $\tau_p = 0.3 \cdot 7.3$ because wind turbines and solar panels generate average power which is, respectively, 2.45 and 7.3 times smaller than their peak power, due to varying wind speed and sun irradiation.

The nominal wind and solar capacities are later distributed between load buses, on which BESs are placed in the considered systems. This is done as follows:

$$\Lambda_m = \frac{P_m^{d,nom}}{\sum_{m \in LB} P_m^{d,nom}} \tag{20}$$

$$P_m^{WT,peak} = \Lambda_m \cdot P^{WT,peak} \tag{21}$$

$$P_m^{PV,peak} = \Lambda_m \cdot P^{PV,peak} \tag{22}$$

where LB is a set of load buses with BESs, on which renewable generators are placed, $m \in LB$, Λ_m is the share of nominal RG capacity to be distributed on bus m, $P_m^{d,nom}$ is the nominal load demand at m-th load bus, $P_m^{WT,peak}$ and $P_m^{PV,peak}$ are the peak power of wind and solar units placed at m-th bus, respectively.

The power output of the wind turbine at the t time step is given below:

$$P_{m,t}^{g,WT} = \begin{cases} \left(\frac{v_{j,t}-v^{in}}{v^{peak}-v^{in}}\right) P_m^{WT,peak} & \text{if } v^{in} \le v_{j,t} < v^{peak} \\ P_m^{WT,peak} & \text{if } v^{peak} \le v_{j,t} \le v^{out} \\ 0 & \text{if } v_{j,t} < v^{in} \text{ or } v_{j,t} > v^{out}. \end{cases} \tag{23}$$

where $v_{j,t}$ is the current wind speed at a given bus. $v^{in}, v^{out}, v^{peak}$ are respectively wind turbine cut-in, cut-out, and peak wind speeds $[ms^{-1}]$. $P_m^{WT,peak}$ is the peak power of the wind turbine, whereas $P_{m,t}^{g,WT}$ is the power produced by the wind turbine at m-th bus and t-th hour.

The peak power output of the PV system can be obtained at an irradiation level of $1000W/m^2$, which is the maximum experienced on Earth. The actual power output of the PV system is proportional to the irradiation and given by (24):

$$P_{m,t}^{g,PV} = \eta \cdot \frac{G_{m,t} \cdot P_m^{PV,peak}}{1000} \tag{24}$$

where $G_{m,t}$ is the current sun irradiance $[Wm^{-2}]$ and $\eta = 0.85$ is the combined efficiency of the solar panel and the converter.

5.2.5 Battery energy storage. In the proposed environment, the number of distributed BES agents that are considered to be controlled in different grid networks are 4, 11, and 37 in IEEE 33, PEGASE 89, and Illinois 200, respectively. The BES units are placed on load buses along with wind turbines and solar panels.

5.3 Environment, observations, actions

This section presents the details of the communication between the agents and the environments in our proposed benchmark. The reward function is a part of the problem definition and was presented in Section 4.1.1.

5.3.1 Environment. In our work, the environment consists of energy grid topology, WT, PV, BESS, and electricity consumers.

5.3.2 States. This work considers $\mathbb{S}$ is a set of environmental states which define the weather and other system conditions such as wind speed $v_{m,t}$, solar irradiation $G_{m,t}$, load demand $\widehat{Pd}_{m,t}$, renewable energy generation $P_{m,t}^{g,WT}$ and $P_{m,t}^{g,PV}$, and current SoC $C_{i,t}$ of each BESS agent.

5.3.3 Observations. The observations $O^i(s_t)$ can be obtained after running the hourly load flow analysis (LFA). The purpose of LFA is to observe total power losses in lines/cables, the surplus or deficit amount of renewable energy in the system represented by power flow at the slack bus P_t^{slack}, voltage limit violation at each bus, and thermal limit/current limit violation of each line.

At the time $t = 1, 2, \ldots$ each i-th agent makes an observation, $o_t^i = O^i(s_t)$, of the environment state s_t. These observations include the following:

(1) System-specific variables, the same for all the agents:
   (a) Day of the year sine and cosine trigonometric functions that represent current progress through the seasons of the year (2 variables),
   (b) Hour of the day sine and cosine trigonometric functions that stand for day-night cycle related observation (2 variables),
   (c) Day of the week one-hot encoding - weekday related trends in energy consumption (7 variables),
   (d) Global net energy forecast - the sum of all energy generation and consumption for different forecast horizons (number of variables equal to the number of forecast horizons),
   (e) Global SoC - overall energy stored in the BESS (1 variable).
(2) Agent-specific variables:
   (a) Local SoC - It is the energy available inside the i-th BESS agent (1 variable per agent),
   (b) Overcharge/overdischarge flag - equals 1 whenever the battery is fully charged, -1 in case of full discharge, and 0 otherwise (1 variable per agent),
   (c) Net regional energy forecast - it is the net energy attributed to a certain agent based on its distance to neighboring buses. The calculation for the distance matrix is discussed in Appendix B (number of variables equal to the number of forecast horizons per agent).

The methodology for creating the regions in the energy grid and observation standardization are presented in Appendices B and C.

5.3.4 Actions. i-th agent action a_t^i belongs to the interval $[-1, 1]$, where -1 denotes full-speed battery discharging, 0 denotes no exchange between the battery and the grid, and 1 denotes full-speed battery charging. Intermediate values denote the proportional speed of battery (dis)charging.

5.3.5 Episode initialization. Before the beginning of each training episode, the SoCs for all batteries are initialized as follows. Firstly, their average is drawn from the uniform distribution, $average \sim U(0, 1)$. Secondly, for each battery, its SoC is drawn from the beta distribution with parameters α, β equal to

$$\langle \alpha, \beta \rangle = \begin{cases} \langle average/(1-average), 1 \rangle & \text{if } average \le 0.5, \\ \langle 1, average/(1-average) \rangle & \text{otherwise.} \end{cases} \tag{25}$$

The expected value of this beta distribution is equal to $average$. For evaluation episodes, we assume initial SoCs equal to 0.5.

6 EXPERIMENTS

6.1 Setup

We have conducted experiments with both SARL and MARL algorithms. For SARL algorithms (A2C, PPO, DDPG, SAC, TD3), we have created a custom training script based on SB3. For multi-agent algorithms, both those where the critic has full observability (MAPPO, MADDPG, MASAC) or is limited to the agent's observations (IPPO, IDDPG, ISAC), we have modified the BenchMARL library to work with our environment wrapped in a PettingZoo interface. The parameters we used for every tested algorithm are available in Appendix D.

The training lasted 1500 2-week episodes sampled randomly from January 2017 to June 2022. The evaluation was done every 25 training episodes on a single one-year episode from July 2022 to June 2023. We report the results from the best evaluation episodes.

Algorithm/System	IEEE 33	PEGASE 89	Illinois 200
A2C	-1577.47 ± 41.06	-2300.12 ± 2.31	-
PPO	-1479.64 ± 12.29	-2317.56 ± 62.55	-2186.11 ± 0.21
DDPG	-1558.28 ± 104.37	-2205.06 ± 54.93	-2188.43 ± 1.92
SAC	-1416.94 ± 17.65	**-2048.70** ± 11.96	**-1957.83** ± 7.91
TD3	-1430.40 ± 15.91	-2111.67 ± 84.28	-2171.86 ± 36.35
MAPPO	-1439.14 ± 36.70	-2364.96 ± 92.92	-2408.29 ± 116.08
IPPO	-	-	-
MADDPG	-1632.83 ± 0.00	-2304.70 ± 2.36	-2312.29 ± 123.28
IDDPG	-1423.79 ± 12.10	-2324.58 ± 2.23	-
MASAC	-1632.78 ± 0.01	-2302.33 ± 0.19	-2200.07 ± 20.91
ISAC	**-1404.41** ± 6.02	-2174.00 ± 25.99	-1977.55 ± 18.95
No batteries	-1633.23	-2296.06	-2181.36

Table 1: Average highest evaluation rewards with standard deviations achieved by every tested algorithm on every system. Dashes denote experiments that failed to produce meaningful results due to instability or lack of convergence on at least one of the runs. A single reward is provided for the no batteries scenario, as it is not dependent on learning.

All experiments were repeated on five random seeds for SARL algorithms and three for MARL algorithms.

For every tested energy grid, we provide information about the reward with no batteries in the system as a point of comparison.

6.2 Results

Results for all tested algorithms and systems are presented in Table 1. In the bottom row of this table, we report the systems' performance with no batteries at all as a reference point for the RL algorithms. It is relatively simple to achieve this reference point. It is enough to produce for all agent actions equal to 0 at all times. Surprisingly, in many cases, the algorithms were only able to reach efficiency of this strategy. In some cases, the tasks proved too difficult to achieve this efficiency.

It is seen that SAC is the best-performing single-agent algorithm, achieving the best results of all tested algorithms on two out of three systems. TD3 also achieved good results, only failing to improve control over no batteries on the largest system. The rest of the SARL methods showed poor performance, slightly improving the results over the no batteries scenarios, failing to do so, or, in the case of the A2C, showing a lack of convergence.

ISAC performs best for multi-agent algorithms, outperforming the no batteries benchmark score on every tested system (the only multi-agent algorithm to do so for PEGASE 89 and Illinois 200 systems). For the IEEE 33 system, ISAC, IDDPG, and MAPPO improved over the no batteries benchmark score, with ISAC being the best algorithm tested. The rest of the algorithm-system combinations failed to improve the control performance over the no batteries scenario, with IPPO failing to converge on two of the systems.

6.3 Discussion

It is visible that larger systems significantly complicate the batteries' control in the power grid. Most tested algorithms improved operations over the no batteries case on the smallest IEEE 33 system. In contrast, most failed to do so on bigger PEGASE 89 and Illinois 200 systems. As the system complexity grows, finding the optimal control becomes more difficult.

We've tested many different neural network configurations for policies and critics for different algorithms. However, this did not affect the results, so we kept the same architecture across all systems and algorithms.

For multi-agent algorithms, those with decentralized critics mostly perform better than those where the critic has full observability. This is especially true for multi-agent SACs, where ISAC achieves good control, while MASAC cannot improve over the scenario of no batteries in every tested system. SAC-based algorithms performed best in single-agent and multi-agent experiments, getting the best results across all systems and being the only methods to improve control on the Illinois 200 system. The superior performance of SAC-based algorithms may be due to how they manage uncertainty. In the presented environments exploration is generally challenging as random charging and discharging the batteries directly translate into energy losses and decrease the rewards. Also, future power net surplus predictions represent important observations available to the agents. The role of the agents boils down to accumulating or disposing of energy in the storage to prepare the system for future deficit or surplus. However, these predictions are burdened with significant noise.

7 CONCLUSIONS

In this paper, we formalized the control of BESS under the non–controllable RES as a Multi-Agent Partially Observable Markov Decision Problem. The proposed benchmark has many practical applications, including comparing existing control methods, developing new control algorithms (both RL-based and non-RL-based), testing these methods across various energy systems, and designing reinforcement learning-maintained energy grids. The research community can broadly use this framework to investigate solutions for efficient EG control further. Moreover, we tested the performance of standard SARL and MARL algorithms in three benchmark environments for reinforcement learning algorithms that simulate the control of a realistic distributed battery energy storage system (BESS). These environments vary in difficulty of the control problem they represent, showing that the fraction of failing algorithms increases with the complexity of the controlled system.

REFERENCES

[1] Ibrahim Althamary, Chih-Wei Huang, and Phone Lin. 2019. A Survey on Multi-Agent Reinforcement Learning Methods for Vehicular Networks. In *2019 15th International Wireless Communications Mobile Computing Conference (IWCMC)*. 1154–1159. https://doi.org/10.1109/IWCMC.2019.8766739

[2] Matteo Bettini, Ryan Kortvelesy, Jan Blumenkamp, and Amanda Prorok. 2022. Vmas: A vectorized multi-agent simulator for collective robot learning. In *International Symposium on Distributed Autonomous Robotic Systems (DARS)*.

[3] Matteo Bettini, Amanda Prorok, and Vincent Moens. 2024. BenchMARL: Benchmarking Multi-Agent Reinforcement Learning. In *Workshop on Aligning Reinforcement Learning Experimentalists and Theorists (ARLET)*.

[4] Madderla Chiranjeevi, Skandha Karlamangal, Tukaram Moger, and Debashisha Jena. 2023. Solar Irradiation Prediction Framework using Regularized Convolutional BiLSTM based Autoencoder Approach. *IEEE Access* (2023).

[5] Abhishek Das, Théophile Gervet, Joshua Romoff, Dhruv Batra, Devi Parikh, Mike Rabbat, and Joelle Pineau. 2019. TarMAC: Targeted multi-agent communication. In *International Conference on Machine Learning (ICML)*. 1538–1546.

[6] Mohammed Dauda and Santosh Panda. 2023. Active and reactive power management in microgrids with high renewable penetration. *Energy Informatics* 6 (2023), 1–22.

[7] Dominik J. Bogucki, Łukasz E. Lepak, Sonam Parashar, Bartłomiej Błachowski, and Paweł Wawrzyński. 2025. EnEnv - Energy Grid Environment for Multi-Agent Reinforcement Learning Benchmarking repository. https://github.com/djbogucki/EnEnv. Accessed: 2025-02-24.

[8] Wei Du and Shifei Ding. 2021. A survey on multi-agent deep reinforcement learning: from the perspective of challenges and applications. *Artificial Intelligence Review* 54 (2021), 3215–3238. Issue 5. https://doi.org/10.1007/s10462-020-09938-y

[9] ECMWF. 2023. Evaluation of ECMWF forecasts, including the 2023 upgrade. https://www.ecmwf.int/en/elibrary/81389-evaluation-ecmwf-forecasts-including-2023-upgrade. Accessed: 2024-15-07.

[10] Benjamin Ellis, Skander Moalla, Mikayel Samvelyan, Mingfei Sun, Anuj Mahajan, Jakob N. Foerster, and Shimon Whiteson. 2022. Smacv2: An improved benchmark for cooperative multiagent reinforcement learning. arXiv2212.07489.

[11] ENTSO-E. 2017. ENTSO-E-Loads: Germany. https://transparency.entsoe.eu/load-domain/r2/totalLoadR2/show. Accessed: 2024-15-07.

[12] Ting-Han Fan, Xian Yeow Lee, and Yubo Wang. 2022. Powergym: A reinforcement learning environment for volt-var control in power distribution systems. In *Learning for Dynamics and Control Conference*. PMLR, 21–33.

[13] Amal Feriani and Ekram Hossain. 2021. Single and Multi-Agent Deep Reinforcement Learning for AI-Enabled Wireless Networks: A Tutorial. *IEEE Communications Surveys Tutorials* 23, 2 (2021), 1226–1252. https://doi.org/10.1109/COMST.2021.3063822

[14] Rezvaneh Golnazari, Saeed Hasanzadeh, Ehsan Heydarian-Forushani, and Innocent Kamwa. 2023. Coordinated active and reactive power management for enhancing PV hosting capacity in distribution networks. *IET Renewable Power Generation* 17 (2023), 345–360.

[15] Xudong Guo, Daming Shi, and Wenhui Fan. 2023. Scalable Communication for Multi-Agent Reinforcement Learning via Transformer-Based Email Mechanism. In *International Joint Conference on Artificial Intelligence (IJCAI)*. arXiv:2301.01919.

[16] Jayesh K. Gupta, Maxim Egorov, and Mykel Kochenderfer. 2017. Cooperative multi-agent control using deep reinforcement learning. In *International conference on autonomous agents and multiagent systems (AAMAS)*. 66–83.

[17] Rajesh Gupta, Gopal Singh, and Vishal Agarwal. 2023. Mitigating Reverse Power Flow in Renewable Energy-Rich Grids. *International Journal of Power Systems Research* 30 (2023). https://doi.org/10.1016/j.ijpsr.2023.06.001

[18] Robin Henry and Damien Ernst. 2021. Gym-ANM: Reinforcement learning environments for active network management tasks in electricity distribution systems. *Energy and AI* 5 (2021), 100092.

[19] Siyi Hu, Yifan Zhong, Minquan Gao, Weixun Wang, Hao Dong, Xiaodan Liang, Zhihui Li, Xiaojun Chang, and Yaodong Yang. 2023. MARLlib: A Scalable and Efficient Multi-agent Reinforcement Learning Library. *Journal of Machine Learning Research* 24, 315 (2023), 1–23. http://jmlr.org/papers/v24/23-0378.html

[20] Woojun Kim, Jongeui Park, and Youngchul Sung. 2021. Communication in multi-agent reinforcement learning: Intention sharing. In *International Conference on Learning Representations (ICLR)*.

[21] Landon Kraemer and Bikramjit Banerjee. 2016. Multi-agent reinforcement learning as a rehearsal for decentralized planning. *Neurocomputing* 190 (2016), 82–94.

[22] Gayathri Krishnamoorthy, Anamika Dubey, and Assefaw H Gebremedhin. 2021. Reinforcement learning for battery energy storage dispatch augmented with model-based optimizer. In *2021 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*. IEEE, 289–294.

[23] Gayathri Krishnamoorthy, Anamika Dubey, and Assefaw H Gebremedhin. 2022. An open-source environment for reinforcement learning in power distribution systems. In *2022 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE, 1–5.

[24] Arjun Kumar and Pritam Sharma. 2023. Control Strategies for Reducing Reverse Power Flow in PV-Integrated Grids. *IEEE Transactions on Sustainable Energy* 14, 1 (2023), 150–162. https://doi.org/10.1109/TSTE.2023.3043321

[25] Karol Kurach, Anton Raichuk, Piotr Stańczyk, Michał Zając, Olivier Bachem, Lasse Espeholt, Carlos Riquelme, Damien Vincent, Marcin Michalski, Olivier Bousquet, and Sylvain Gelly. 2020. Google research football: A novel reinforcement learning environment. In *AAAI Conference on Artificial Intelligence (AAAI)*.

[26] Joel Z. Leibo, Edgar Dué nez Guzmán, Alexander Sasha Vezhnevets, John P. Agapiou, Peter Sunehag, Raphael Koster, Jayd Matyas, Charles Beattie, Igor Mordatch, and Thore Graepel. 2021. Scalable evaluation of multi-agent reinforcement learning with melting pot. arXiv2107.06857.

[27] Dapeng Li, Hang Dong, Lu Wang, Bo Qiao, Si Qin, Qingwei Lin, Dongmei Zhang, Qi Zhang, Zhiwei Xu, Bin Zhang, and Guoliang Fan. 2024. Verco: Learning Coordinated Verbal Communication for Multi-agent Reinforcement Learning. https://arxiv.org/abs/2404.17780 arXiv:2404.17780.

[28] Tianxu Li, Kun Zhu, Nguyen Cong Luong, Dusit Niyato, Qihui Wu, Yang Zhang, and Bing Chen. 2022. Applications of Multi-Agent Reinforcement Learning in Future Internet: A Comprehensive Survey. *IEEE Communications Surveys Tutorials* 24, 2 (2022), 1240–1279. https://doi.org/10.1109/COMST.2022.3160697

[29] Yanzhong Liu, Zhi Jiang, Zhaohua Xing, Lijuan Hao, and Boyang Qu. 2023. Economic and low-carbon island operation scheduling strategy for microgrid with renewable energy. *Energy Reports* 8 (2023), 196–204. https://doi.org/10.1016/j.egyr.2022.12.007

[30] Yong Liu, Weixun Wang, Yujing Hu, Jianye Hao, Xingguo Chen, and Yang Gao. 2020. Multi-agent game abstraction via graph attention neural network. In *AAAI Conference on Artificial Intelligence (AAAI)*. 7211–7218.

[31] Zeyang Liu, Lipeng Wan, Xue Sui, Zhuoran Chen, Kewu Sun, and Xuguang Lan. 2023. Deep Hierarchical Communication Graph in Multi-Agent Reinforcement Learning. In *International Joint Conference on Artificial Intelligence (IJCAI)*. 208–216. https://doi.org/10.24963/ijcai.2023/24

[32] Zichuan Liu, Yuanyang Zhu, and Chunlin Chen. 2023. NA^2Q: Neural Attention Additive Model for Interpretable Multi-Agent Q-Learning. In *International Conference on Machine Learning (ICML)*. 22539–22558.

[33] Ryan Lowe, Yi I. Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. 2017. Multi-agent actor-critic for mixed cooperative-competitive environments. In *Neural Information Processing Systems (NIPS)*, Vol. 30.

[34] Xiping Ma, Chen Liang, Xiaoyang Dong, and Yaxin Li. 2023. Multi-objective reactive power optimization strategy of power system considering large-scale renewable integration. *Frontiers in Energy Research* 11 (2023), 102–120.

[35] Subham Misra, Pramod Kumar Panigrahi, and Banshidhar Dey. 2023. An efficient way to schedule dispersed generators for a microgrid system's economical operation under various power market conditions and grid involvement. *International Journal of Systems Assurance Engineering and Management* 14 (2023), 1–12. https://doi.org/10.1007/s13198-023-01983-4

[36] Muhammad Saeed Nazir, Fahd M. Almasoudi, Ahmed N. Abdalla, Cheng Zhu, Sherif S. Khaled, and Faisal Alatawi. 2023. Multi-objective optimal dispatching of combined cooling, heating and power using hybrid gravitational search algorithm and random forest regression: towards the microgrid orientation. *Energy Reports* 9 (2023), 1926–1936. https://doi.org/10.1016/j.egyr.2023.03.028

[37] Komail Nekooei, Malihe M Farsangi, Hossein Nezamabadi-Pour, and Kwang Y Lee. 2013. An improved multi-objective harmony search for optimal placement of DGs in distribution systems. *IEEE Transactions on smart grid* 4, 1 (2013), 557–567.

[38] Thanh Thi Nguyen, Ngoc Duy Nguyen, and Saeid Nahavandi. 2020. Deep Reinforcement Learning for Multiagent Systems: A Review of Challenges, Solutions, and Applications. *IEEE Transactions on Cybernetics* 50, 9 (2020), 3826–3839. https://doi.org/10.1109/TCYB.2020.2977374

[39] Zepeng Ning and Lihua Xie. 2024. A survey on multi-agent reinforcement learning and its application. *Journal of Automation and Intelligence* 3, 2 (2024), 73–91. https://doi.org/10.1016/j.jai.2024.02.003

[40] Afshin Oroojlooy and Davood Hajinezhad. 2023. A review of cooperative multi-agent deep reinforcement learning. *Applied Intelligence* 53 (2023), 13677–13722. Issue 11. https://doi.org/10.1007/s10489-022-04105-y

[41] James Orr and Ayan Dutta. 2023. Multi-Agent Deep Reinforcement Learning for Multi-Robot Applications: A Survey. *Sensors* 23, 7 (2023). https://doi.org/10.3390/s23073625

[42] Georgios Papoudakis, Filippos Christianos, Lukas Schäfer, and Stefano V. Albrecht. 2021. Benchmarking Multi-Agent Deep Reinforcement Learning Algorithms in Cooperative Tasks. In *Neural Information Processing Systems (NeurIPS)*.

[43] Yansong Pei, Yiyun Yao, Junbo Zhao, Fei Ding, and Jiyu Wang. 2023. Two-Stage Deep Reinforcement Learning for Distribution System Voltage Regulation and Peak Demand Management. In *2023 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE, 1–5.

[44] Bei Peng, Tabish Rashid, Christian A. Schroeder de Witt, Pierre-Alexandre Kamienny, Philip H. S. Torr, Wendelin Böhmer, and Shimon Whiteson. 2021. Facmac: Factored multi-agent centralised policy gradients. In *Neural Information Processing Systems (NIPS)*.

[45] Tabish Rashid, Gregory Farquhar, Bei Peng, and Shimon Whiteson. 2020. Weighted QMIX: Expanding Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. In *Advances in Neural Information Processing Systems (NeurIPS)*. 1–20.

[46] Tabish Rashid, Mikayel Samvelyan, Christian Schroeder de Witt, Gregory Farquhar, Jakob Foerster, and Shimon Whiteson. 2018. QMIX: Monotonic Value Function Factorisation for Deep Multi-Agent Reinforcement Learning. In *International Conference on Machine Learning (ICML)*. 1–14.

[47] Cephas Samende, Zhong Fan, Jun Cao, Renzo Fabián, Gregory N Baltas, and Pedro Rodriguez. 2023. Battery and hydrogen energy storage control in a smart energy network with flexible energy demand using deep reinforcement learning. *Energies* 16, 19 (2023), 6770.

[48] Mikayel Samvelyan, Tabish Rashid, Christian Schroeder de Witt, Gregory Farquhar, Nantas Nardelli, Tim G. J. Rudner, Chia-Man Hung, Philip H. S. Torr, Jakob Foerster, and Shimon Whiteson. 2019. The starcraft multi-agent challenge. In *International conference on autonomous agents and multiagent systems (AAMAS)*.

[49] Lukas M. Schmidt, Johanna Brosig, Axel Plinge, Bjoern M. Eskofier, and Christopher Mutschler. 2022. An Introduction to Multi-Agent Reinforcement Learning and Review of its Application to Autonomous Mobility. In *International Conference on Intelligent Transportation Systems (ITSC)*. 1342–1349. https://doi.org/10.1109/ITSC55140.2022.9922205

[50] Alaa Selim, Huadong Mo, Hemanshu Pota, and Daoyi Dong. 2023. Optimal Scheduling of Battery Energy Storage Systems Using a Reinforcement Learning-Based Approach. *IFAC-PapersOnLine* 56, 2 (2023), 11741–11747.

[51] Kyunghwan Son, Daewoo Kim, Wan Ju Kang, David Earl Hostallero, and Yung Yi. 2019. QTRAN: Learning to Factorize with Transformation for Cooperative Multi-Agent Reinforcement Learning. In *International Conference on Machine Learning (ICML)*. 1–18.

[52] Sainbayar Sukhbaatar, Arthur Szlam, and Rob Fergus. 2016. Learning multiagent communication with backpropagation. In *Advances in Neural Information Processing Systems (NIPS)*, Vol. 29.

[53] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z. Leibo, Karl Tuyls, and Thore Graepel. 2018. Value-Decomposition Networks for Cooperative Multi-Agent Learning Based on Team Reward. In *International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 2085–2087.

[54] Richard S. Sutton and Andrew G. Barto. 2018. *Reinforcement Learning: An Introduction* (second ed.). The MIT Press.

[55] Weihao Tan, Wentao Zhang, Shanqi Liu, Longtao Zheng, Xinrun Wang, and Bo An. 2024. True knowledge comes from practice: Aligning large language models with embodied environments via reinforcement learning. In *International Conference on Learning Representations (ICLR)*.

[56] Jianhao Wang, Zhizhou Ren, Terry Liu, Yang Yu, and Chongjie Zhang. 2021. QPLEX: Duplex Dueling Multi-Agent Q-Learning. In *International Conference on Learning Representations (ICLR)*. 1–27.

[57] Tonghan Wang, Jianhao Wang, Chongyi Zheng, and Chongjie Zhang. 2019. Learning nearly decomposable value functions via communication minimization.

[58] Ermo Wei, Drew Wicke, David Freelan, and Sean Luke. 2018. Multiagent Soft Q-Learning. In *AAAI Conference on Artificial Intelligence (AAAI)*. 1–7.

[59] Christian Schroeder De Witt, Tarun Gupta, Denys Makoviichuk, Viktor Makoviychuk, Philip HS Torr, Mingfei Sun, and Shimon Whiteson. 2020. Is independent learning all you need in the starcraft multi-agent challenge? arXiv:2011.09533.

[60] Annie Wong, Thomas Bäck, Anna V. Kononova, and Aske Plaat. 2023. Deep multiagent reinforcement learning: challenges and directions. *Artificial Intelligence Review* 56 (2023), 5023–5056. Issue 6. https://doi.org/10.1007/s10462-022-10299-x

[61] Pamul Yadav, Ashutosh Mishra, and Shiho Kim. 2023. A Comprehensive Survey on Multi-Agent Reinforcement Learning for Connected and Automated Vehicles. *Sensors* 23, 10 (2023). https://doi.org/10.3390/s23104710

[62] Yaodong Yang, Jianye Hao, Ben Liao, Kun Shao, Guangyong Chen, Wulong Liu, and Hongyao Tang. [n.d.]. Qatten: A general framework for cooperative multiagent reinforcement learning. arXiv:2002.03939.

[63] Chao Yu, Akash Velu, Eugene Vinitsky, Jiaxuan Gao, Yu Wang, Alexandre Bayen, and Yi Wu. 2022. The Surprising Effectiveness of PPO in Cooperative Multi-Agent Games. In *Neural Information Processing Systems (NeurIPS)*. 1–30.

[64] Lei Yuan, Jianhao Wang, Fuxiang Zhang, Chenghe Wang, Zongzhang Zhang, Yang Yu, and Chongjie Zhang. 2022. Multi-agent incentive communication via decentralized teammate modeling. In *AAAI Conference on Artificial Intelligence (AAAI)*. 9466–9474.

[65] Lianmin Zheng, Jiacheng Yang, Han Cai, Weinan Zhang, Jun Wang, and Yong Yu. 2018. Magent: A many-agent reinforcement learning platform for artificial collective intelligence. In *AAAI Conference on Artificial Intelligence (AAAI)*.

[66] Qi Zhou, Hong Wang, and Zhihong Lin. 2023. Impact of Reverse Power Flow on Grid Stability in High Penetration PV Systems. *Journal of Renewable Energy Integration* 25 (2023), 87–98. https://doi.org/10.1016/j.jrei.2023.05.009