Networked Multiagent Reinforcement Learning for Peer-to-Peer Energy Trading

Chen Feng, and Andrew L. Liu

***Abstract*—Utilizing distributed renewable and energy storage resources in local distribution networks via peer-to-peer (P2P) energy trading has long been touted as a solution to improve energy systems' resilience and sustainability. Consumers and prosumers (those who have energy generation resources), however, do not have the expertise to engage in repeated P2P trading, and the zero-marginal costs of renewables present challenges in determining fair market prices. To address these issues, we propose multi-agent reinforcement learning (MARL) frameworks to help automate consumers' bidding and management of their solar PV and energy storage resources, under a specific P2P clearing mechanism that utilizes the so-called supply-demand ratio. In addition, we show how the MARL frameworks can integrate physical network constraints to realize voltage control, hence ensuring physical feasibility of the P2P energy trading and paving way for real-world implementations.**

I. Introduction

As our society strives to transition towards sustainable energy sources, distributed renewable resources and energy storage are increasingly seen as key components in creating resilient and sustainable energy systems. Peer-to-peer (P2P) energy trading in local energy markets offers a promising approach to decentralize and optimize the allocation of distributed energy resources (DERs). This model not only empowers consumers and those who can store or generate energy, known as prosumers, to trade energy directly but also promotes renewable energy usage and reduces energy losses.

However, implementing P2P trading presents significant challenges. First, consumers and prosumers lack the technical expertise required to engage in repeated P2P trading and manage their energy resources efficiently. Second, renewable energy, such as solar, often has zero marginal costs, which introduces difficulties in determining fair market prices for energy trades: a uniform-pricing auction resembling a wholesale energy market would not work since the market clearing prices would be zero most of the time. Third, while P2P trading only entails financial transactions, the delivery of energy occurs over the physical distribution networks. How to maintain network feasibility in a local P2P trading market is an important yet open question.

In response to these challenges, this paper introduces a decentralized framework utilizing multiagent reinforcement learning (MARL), which is designed to automate the bidding processes of agents, enhancing the management efficiency of their solar PV and energy storage resources. Additionally, the framework ensures the feasibility of a distribution network. Specifically, inspired by the algorithm in [1], we propose a consensus-based actor-critic algorithm for a decentralized MARL, in which each agent in a repeated P2P auction is modeled as a Markov decision problem (MDP). By allowing agents to exchange information and reach agreements, the proposed framework facilitates efficient decision-making and resource allocation, while mitigating the computational and privacy challenges associated with certain centralized learning approaches. In addition, the shared network constraints, such as voltage regulation, can be learned through the decentralized approach with a constraint violation incurring a (fictitious) penalty to each agent's reward function. We consider this development a significant step towards practical real-world implementations of P2P energy trading.

Chen Feng is with School of Industrial Engineering, Purdue University, West Lafayette, IN, USA, email: feng219@purdue.edu.

Andrew L. Lu is with School of Industrial Engineering, Purdue University, West Lafayette, IN, USA, email: andrewliu@purdue.edu.

In addition to the modeling and algorithm development, we establish the theoretical conditions for the consensus-MARL algorithm to converge to an asymptotically stable equilibrium. We consider this a significant advantage over a purely decentralized approach without agents' communication, such as the one in our previous work [2], in which each agent uses the proximal policy optimization (PPO) approach [3] to solve their own MDP and ignores multiagent interaction.

Through numerical simulations, we compare three different frameworks to implement P2P trading while considering network constraints: purely decentralized trading using PPO, consensus-MARL, and a centralized learning and decentralized execution approach using the multiagent deep deterministic policy gradient (MADDPG) algorithm [4]. Our findings indicate that consensus-MARL achieves higher average rewards compared to both PPO and MADDPG.

The rest of the paper is organized as follows. Section II reviews the literature on P2P energy trading and emphasizes the contribution of our work. Section III introduces the market clearing mechanism based on the supply-demand ratio, first proposed in [5], which can avoid zero-clearing prices when the supply resources have zero marginal costs. Section IV provides the detailed consensus-MARL formulation and algorithm, as well as establishing convergence results. The specific setups and data of our simulation are described in Section V, along with numerical results. Finally, Section VI summarizes our work and points out several future research directions.

II. Literature Review

The literature on decentralized control of energy transactions in a distribution network is extensive and diverse. Broadly speaking, there are two predominant approaches: the distributed optimization approach and P2P bilateral trading. A key method in the former approach is the alternating direction multiplier method (ADMM), which solves centralized dispatch problems by iteratively updating price signals for consumer/prosumer optimization (such as [6]–[8]). Despite ADMM's theoretical convergence under convexity, practical challenges arise: (i) As an algorithmic approach rather than a market design, ADMM is subject to the same difficulties faced by a uniform-pricing-based market of all zero-marginal-cost resources. Specifically, it is prone to 'bang-bang' pricing: market clearing prices plummet to zero in oversupply or spike to a price ceiling in undersupply. Addressing this with additional market mechanisms, such as adding ancillary service markets, akin to wholesale markets, could overly complicate distribution-level markets for energy trading. (ii) Most distributed algorithms, including ADMM, are static, lacking real-time adaptability. While online ADMM variants exist [9], ensuring network constraints within each iteration remains an open question. (iii) Assuming consumers or prosumers can solve complex optimization problems is often unrealistic due to potential expertise or resource limitations.

These limitations form the basis of our interest in exploring P2P bilateral trading. Consequently, our literature review will focus on this alternative approach. There have been multiple review papers on the large and ever-growing literature on P2P energy trading [10]–[13]. Within the literature, we focus on the works that explicitly consider distribution network constraints. [14] proposes a continuous double auction framework within which physical network constraints are maintained through a sensitivity-based approach. The agents in the auction are the so-called zero-intelligence-plus (ZIP) traders, who employ simple adaptive mechanisms without any learning of the repeated multi-agent interactions. More sophisticated than ZIP, a fully decentralized MARL is adopted in [15] to realize decentralized voltage control and is extended in [2] to form a P2P energy trading market with physical constraints. These two papers implement a completely decentralized approach in the sense that each agent ignores the multi-agent interaction as well and just learns their single-agent policy to maximize their own payoff.

Another MARL approach that has been explored is the centralized training and decentralized execution (CTDE) framework, of which the multi-agent deep deterministic policy gradient (MADDPG) method [4] is a prominent example. It is applied in [16] to solve the autonomous voltage control problem In [17], a modified MADDPG algorithm is proposed for a double-sided auction market to facilitate P2P energy trading, although physical constraints are not considered in their model. While the CTDE framework can lead to more efficient learning and robust outcomes, its dependence on a centralized entity for collecting and coordinating all agents' data introduces difficulties for practical implementations and potential vulnerabilities, such as increased risk of single-point failures and higher susceptibility to cyber-attacks

To overcome the drawbacks of fully decentralized and CTDE-based MARL, algorithms with communication among agents have also been developed. A notable contribution in this area is the consensus-based MARL algorithm designed for discrete-space problems, as presented in [18], and its extension to continuous-space problems in [1]. This model allows agents to learn and implement policies in a decentralized manner with limited communication, underpinned by a theoretical guarantee of convergence under certain conditions. Separately, [19] introduced a Scalable Actor Critic (SAC) framework, particularly relevant for large-scale networks where an individual agent's state transition is influenced only by its neighbors. SAC is distinguished by its reduced communication needs compared to the CTDE framework and offers a theoretical guarantee for its policy convergence as well. However, the SAC framework is only established for discrete state space problems, and its extension and effectiveness in continuous-state space applications remain unexplored.

In this work, we present several key contributions: First, we develop a comprehensive framework for a P2P energy trading market while integrating a consensus-based MARL algorithm. This framework facilitates automated control and bidding of DERs while allowing for decentralized learning of voltage constraints. Second, we present the sufficient conditions required for the convergence of the MARL algorithm and show the details of implementing this algorithm using deep neural networks. Third, we conduct a comparative analysis of market outcomes derived from three distinct MARL strategies: a naive decentralization approach, the MADDPG approach, and the proposed consensus-based MARL. Our findings reveal that the consensus-based approach notably outperforms the others in achieving better market outcomes.

III. Market Clearing Mechanism

In this section, we first describe a clearing mechanism, referred to as the supply-demand-ratio (SDR) [5], that can address the two potential issues faced by a renewable-dominated P2P market: (i) marginal-cost-based pricing yielding zero market prices, and (ii) the bang-bang outcomes in a double-auction-based mechanism, as the reserve prices for buyers (the utility rates) and the sellers (usually the feed-in tariffs) are publicly known. The second issue is unique to the P2P energy market, in our opinion, and the bang-bang phenomenon is documented in [20] and will be explained further in this section.

The SDR-clearing mechanism in P2P markets has been detailed in our earlier work [2]. For completeness, we briefly revisit it here. Let $\mathcal{I} = \{1, 2, ..., I\}$ represent all consumers and prosumers, all referred to as agents. We assume that trading among the agents takes place in fixed rounds (such as hourly or every 15 minutes) and is cleared ahead of time (such as day-ahead or hour-ahead). In each trading round t, each agent $i \in \mathcal{I}$ submits bids ($b_{i,t}$) to buy (if $b_{i,t} < 0$) or sell (if $b_{i,t} > 0$) energy. Agents can switch roles between buyers and sellers across different rounds, but not within the same round. For a round t, the sets of buyers and sellers are denoted as $\mathcal{B}_t = \{i : b_{i,t} < 0\}$ and $\mathcal{S}_t = \{i : b_{i,t} \geq 0\}$, respectively.

The SDR mechanism, originally proposed in [5], is an approach to clear a market with all zero-marginal-cost supply resources, which is straightforward to implement. An SDR is defined as the ratio between total supply and demand bids at t; that is,

$$SDR_t := \sum_{i \in \mathcal{S}_t} b_{i,t} \Big/ - \sum_{i \in \mathcal{B}_t} b_{i,t}. \tag{1}$$

To ensure that SDR_t is well-defined, we assume $\mathcal{B}_t \neq \emptyset$ for any t, a reasonable assumption given agents' continuous need for energy. In rare cases where every agent is a prosumer with excess energy at time t, the P2P market can be temporarily suspended, and surplus energy sold to the grid at a predefined price, which will not impact our model or algorithm framework. Conversely, if $\mathcal{S}_t$ is $\emptyset$, indicating no excess energy for sale at t, then $SDR_t = 0$.

Based on its definition, when $SDR_t > 1$, it means that there is an over-supply. In this case, we assume that the excess energy in the P2P market is sold to a utility company, a distribution system operator (DSO), or an aggregator at a pre-defined rate, generally denoted as FIT (feed-in tariff).[1] If $0 \leq SDR_t < 1$, indicating over-demand in t, then the unmet demand bids will purchase energy at a pre-defined utility rate, denoted as UR. Without loss of generality, we assume that $FIT < UR$.[2]

The SDR mechanism determines the market clearing price as follows:

$$\begin{aligned} P_t &:= P(SDR_t) \\ &:= \begin{cases} (FIT - UR) \cdot SDR_t + UR, & 0 \leq SDR_t \leq 1; \\ FIT, & SDR_t > 1. \end{cases} \end{aligned} \tag{2}$$

In (2), P_t denotes the market clearing price in round t, which is a piece-wise linear function with respect to the SDR_t, as illustrated in Figure 1.

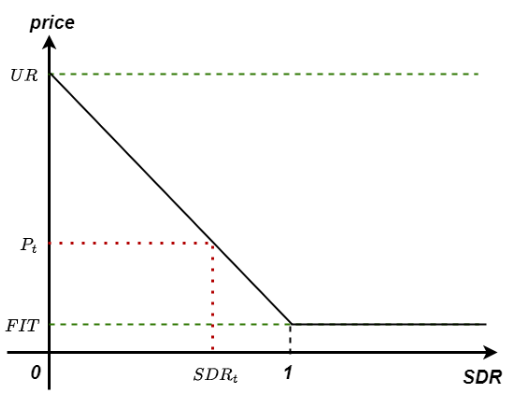

Figure 1: Market clearing using SDR

While the SDR approach may be criticized for not being based on economic theories, we argue that from our perspectives, the simplicity and transparency of the SDR mechanism outweigh the criticisms, and it is well-suited for a P2P energy market with all-zero marginal cost resources. In our previous work [20], it is identified that a P2P energy market using traditional double-sided auction may not work well due to the special characteristics of such a market. Since uncleared electricity demand bids have to be bought at the rate of UR, and uncleared energy supply bids have to be sold at FIT, with the UR and FIT known to all market participants. With all zero-marginal cost resources, agents would not know how to price their bids in a marginal-cost-based clearing mechanism, and the public information of the UR and the FIT naturally provides the focal points for sellers to bid at UR and buyers to bid at FIT, aiming to maximize their own payoffs. This would lead to a bang-bang type market outcome: when there is overdemand, the market price would be UR; reversely, when there is oversupply, the price would be FIT. We view the SDR approach as a fair and reasonable alternative to the marginal-cost-based approach to avoid such bang-bang outcomes, especially for a new market in which the supply resources are likely much less than demand.

Additionally, the SDR method offers two key benefits: First, it maintains the market clearing price between UR and FIT, ensuring that transactions for buyers and sellers are at least as favorable as dealing directly with the utility. Second, it simplifies the bidding process by requiring only quantity bids, avoiding the complexity of submitting bids in price-quantity pairs.

IV. Decentralized Learning in P2P Trading with Consensus Updates

In this section, we first formulate the decision-making problem faced by each agent as a MDP. (For brevity, we only describe a prosumer's problem. A pure consumer's problem is similar, just without the supply bids.) We then introduce the details of the consensus-MARL algorithm.

A. Markov Decision Process for a Prosumer

In our setup, we assume that each prosumer locates at one bus of the distribution network and has two resources under their control: a solar photovoltaic (PV) system with a smart inverter, and a rechargeable battery as energy storage. Each prosumer also has a fixed baseload of energy consumption. At a particular time period t, whether a prosumer is a net energy seller or a buyer depends on their baseload versus the level of PV generation and energy storage.

Due to the presence of energy storage, each agent's bidding and charging decisions are linked over time, which are naturally modeled through dynamic programming. Since an agent's decisions at t only depend on its energy storage level at $t-1$ and the PV generation in t, a discrete-time MDP model is suitable here. In the following, we describe the key building blocks of each agent's MDP model.

Observation and state variables: Agent $i \in \mathcal{I}$ at time step t is assumed to receive the system states $s_t = (s_{1,t}, \ldots, s_{I,t}) \in \mathcal{S} := \Pi_{i=1}^{I} \mathcal{S}_i$, which is the concatenation of the states of each agent, where I is the total number of agents. Specifically, the individual state variable is defined as $s_{i,t} := (d_{i,t}^p, d_{i,t}^q, e_{i,t}, PV_{i,t}) \in \mathcal{S}_i$, where $d_{i,t}^p$ and $d_{i,t}^q$ are the inflexible demand of active and reactive power of agent i, $e_{i,t}$ is the state of charge of agent i's energy storage system, and $PV_{i,t}$ is the PV (real) power generation in t.

Among the state variables, $(d_{i,t}^p, d_{i,t}^q, PV_{i,t})$ are only observations because their transitions adhere to underlying distributions that are independent of agents' actions. In contrast, the

[1] The FIT rate is a rate set by utilities/policymakers. It can be set as zero and will not affect the pricing mechanism in any way.

[2] If $FIT > UR$, it means that energy consumers pay more than the utility rate to purchase energy from the prosumers, which is equivalent to a direct subsidy from energy consumers (including low-income consumers who do not have the means to invest in DERs) to prosumers. This cannot be justified from an equity perspective.

transition of $e_{i,t}$ is influenced by agents' actions, with details to be elaborated further below after agents' actions are defined.

The assumption that each agent is required to observe the whole system's state may appear to be strong in a decentralized setting and is prone to criticisms and privacy concerns. Our responses are twofold: first, the identity of the agents is irrelevant, and hence, from an agent's perspective, which state variable corresponds to which specific agent is not known. Second, the proposed algorithms are intended to be implemented on devices that can receive information from the grid (generally referring to a utility or a distributed system operator) and automatically participate in the repeated P2P market without human intervention (aka control automation). The information received by the devices should be encrypted and can only be decrypted by the control-automation devices. Granted that the devices could be compromised, there have been works on how to address adversarial attacks in a multi-agent reinforcement learning framework [21], which leads to interesting future research directions.

Action or control variables: at time t, the actions that agent i can take are represented by a vector $a_{i,t} := (a^q_{i,t}, a^e_{i,t}) \in \mathcal{A}_i = \mathcal{A}^q_i \times \mathcal{A}^e_i$, where $a^q_{i,t}$ is the reactive power injected or absorbed by agent i's smart inverter, and $a^e_{i,t}$ is energy charged (if $a^e_{i,t} > 0$) to or discharged (if $a^e_{i,t} < 0$) from the battery. The feasible action spaces of $a^q_{i,t}$ and $a^e_{i,t}$ are generically denoted by $\mathcal{A}^q_i$ and $\mathcal{A}^b_i$, respectively. In the specific context of our model, it is safe to assume that the action space $\mathcal{A}_i = \mathcal{A}^q_i \times \mathcal{A}^e_i$ is bounded.

State transition: Out of the state variables defined earlier, the energy storage level $e_{i,t}$ is the only one that is directly affected by an agent's own action. The state transition for $e_{i,t}$ can be written as follows:

$$e_{i,t+1} = \max\left\{ \min\left[e_{i,t} + \eta^c_i \max(a^e_{i,t}, 0) + \frac{1}{\eta^d_i}\min(a^e_{i,t}, 0), \bar{e}_i\right], 0\right\}, \tag{3}$$

where η^c_i and η^d_i are the charging and discharging efficiency of agent i's battery, respectively, and $\bar{e}_i$ is the battery capacity. The charging efficiency represents the ratio of the amount of energy effectively stored in the system to the energy input during the charging process; while the discharging efficiency is the ratio of the energy delivered by the system during discharge to the energy that was initially stored in it. Their product, $\eta^c_i \times \eta^d_i$, yields the so-called round-trip efficiency. The outside "max" operation in (3) is to ensure that the energy level in the battery will not be negative; while the first "min" operation in the bracket is to ensure that the battery storage capacity is not exceeded.

Reward: Each agent i's reward function is affected by both the collective actions of the agents and the system states. Specifically, agent i's reward in time step t, denoted by $r_{i,t}$, is a random variable whose conditional expectation has two components as follows:

$$\begin{aligned}\mathbb{E}[r_{i,t}|s_t, a_t] &:= R_i(a_{i,t}; a_{-i,t}, s_t) \\ &= R^m_i(a^e_{i,t}; a^e_{-i,t}, s_t) + R^v(a_{i,t}; a_{-i,t}, s_t)/I.\end{aligned} \tag{4}$$

In (4), the notation $a_{-i,t}$ refers to the collection of all other agents' actions, excluding agent i's; while $a^e_{-i,t}$ refers to only the energy charge/discharge decisions of the other agents. The first term R^m_i in (4) is the energy purchase cost or sales profit of agent i from the P2P energy market (with the superscript 'm' standing for 'market'), whose formulation is as follows:

$$R^m_i := \begin{cases} \mathbb{I}_{i\in\mathcal{B}_t} \times \Big[SDR_t \cdot P_t \cdot b_{i,t} + \ (1 - SDR_t) \cdot UR \cdot b_{i,t}\Big] \\ + \ \mathbb{I}_{i\in\mathcal{S}_t} \times \Big(P_t \cdot b_{i,t}\Big), \quad \text{if } 0 \le SDR_t \le 1, \\ \\ FIT \cdot b_{i,t}, \quad \text{if } SDR_t > 1, \end{cases} \tag{5}$$

where $\mathbb{I}_{i\in\mathcal{B}_t}$ is an indicator function such that $\mathbb{I}_{i\in\mathcal{B}_t} = 1$ if $i \in \mathcal{B}_t$, and 0 otherwise. Similarly, $\mathbb{I}_{i\in\mathcal{S}_t} = 1$ if $\in \mathcal{S}_t$ and 0 otherwise. SDR_t is the supply-demand-ratio as defined in (1), which depends on all agents' bids, and the bids further depend on each agent's charging/discharging decisions, as well as the PV generation and the baseload. Specifically, the bids can be explicitly written as

$$b_{i,t} = \begin{cases} PV_{i,t} - d^p_{i,t} - \min(a^e_{i,t}, \frac{\bar{e}_i - e_{i,t}}{\eta^c_i}), & \text{if } a^e_{i,t} \ge 0, \\ PV_{i,t} - d^p_{i,t} - \max(a^e_{i,t}, -e_{i,t} \cdot \eta^d_i), & \text{otherwise}, \end{cases} \tag{6}$$

which is simply the net energy of PV generation minus baseload demand (of real power) and charge to the battery. The market clearing price P_t in (5) is defined in (2).

Note that when $SDR_t < 1$, not all demand bids will be cleared in the P2P market at time t. In such instances, they will need to purchase the needed energy at the utility rate, leading to the $UR \cdot b_{i,t}$ term when defining the reward R^m_i in equation (5). To ensure fairness and prevent any inadvertent favoritism among the demand bids, we employ a mechanism where each demand bid is partially cleared. Specifically, the clearance proportion is exactly SDR_t (which represents the percentage of total demand bids cleared), and the uncleared bid for every agent is then $(1 - SDR_t) \cdot b_{i,t}$. This approach of proportional scaling echoes the rules used in multi-unit double auctions in [22].

The second term in the reward function, R^v, is the system voltage violation penalty at a given time t. To provide an explicit form of R^v, we first write out the standard bus-injection model. Assume that the network has N buses, we have that for $\kappa = 1, \ldots, N$ (the time index t is omitted below for simplicity):

$$\begin{aligned} p_\kappa &= \sum_{j=1}^N |V_\kappa||V_j|(G_{\kappa j}\cos(\alpha_\kappa - \alpha_j) + B_{\kappa j}\sin(\alpha_\kappa - \alpha_j)), \\ q_\kappa &= \sum_{j=1}^N |V_\kappa||V_j|(G_{\kappa j}\sin(\alpha_\kappa - \alpha_j) - B_{\kappa j}\cos(\alpha_\kappa - \alpha_j)). \end{aligned} \tag{7}$$

In the above system of nonlinear equations, p_κ's and q_κ's are input data, with p_κ being the net real power injection (or withdrawal) at bus κ (i.e. $p_\kappa = \sum_{i\in\mathcal{N}_\kappa} b_i$, with b_i given in (6) and $\mathcal{N}_\kappa$ denoting the set of agents at bus κ), and q_k being the net reactive power flow at bus k (i.e. $q_\kappa = \sum_{i\in\mathcal{N}_\kappa}(-d^q_i + a^q_i)$). The variables are $|V_\kappa|$ and α_κ for $\kappa = 1, \ldots, N$, with $|V_\kappa|$

being the voltage magnitude and α_κ the phase angle at bus κ. The terms $G_{\kappa j}$ and $B_{\kappa j}$ are parameters that represent the real and imaginary parts of the admittance of branch κ-j in the network, respectively.

At a given round t of the P2P trading, for a given set of the real and reactive power injection/withdrawal as the result of agents' bidding, if the system of equations (7) has a solution,[3] then we can define the voltage violation penalty as follows:

$$R^v := -\lambda \sum_{\kappa=1}^{N} \text{clip}\Big[\max(|V_{\kappa,t}| - \overline{V}_\kappa, \underline{V}_\kappa - |V_{\kappa,t}|), \{0, M\}\Big]. \tag{8}$$

In (8), $\overline{V}_\kappa$ and $\underline{V}_\kappa$ represent the upper and lower limit of bus k's voltage magnitude, respectively. Let f_{clip} denote a generic 'clip' function in the form of $f_{clip} = \text{clip}[x, \{v_{min}, v_{max}\}]$, which is defined as $f_{clip} = x$ when $v_{min} \le x \le v_{max}$, $f_{clip} = v_{min}$ if $x < v_{min}$, and $f_{clip} = v_{max}$ if $x > v_{max}$. Based on the clip function definition, if the voltage magnitudes $|V_{\kappa,t}|$ obtained from solving equations (7) are within the voltage limit of $[\underline{V}_\kappa, \overline{V}_\kappa]$ on every bus $\kappa = 1, \ldots, N$, then $R^v = 0$; if there is voltage violation at a certain bus κ, the term $\max(|V_{\kappa,t}| - \overline{V}_\kappa,\ \underline{V}_\kappa - |V_{\kappa,t}|)$ becomes positive, and R^v becomes a negative number with an arbitrary positive number λ to amplify the voltage violation. The other positive parameter in (7), M, is just to ensure that the reward R^v is bounded below, a technical condition needed to ensure the convergence of the consensus MARL algorithm to be introduced in the next section.

If (7) does not have a solution with a given set of agents' bids, we can simply set $R^v = -\lambda \cdot N \cdot M$. We want to emphasize that the voltage violation does not result in actual financial penalties for the agents since the market clearing happens before real-time delivery. It is simply feedback sent to agents for them to train their reinforcement learning policies.

After market-clearing and solving for voltage magnitudes, the total reward for agent i at time step t is realized, as defined in (4). Note that while we do not explicitly assume to have a DSO in the distribution network, an entity is still needed to solve Equation (7). A utility company can assume such a role. The solution process may even be carried out by a distributed ledger system, such as on a blockchain.

Objective function: At each t, agent i receives the system state s_t, chooeses an action $a_{i,t} \in \mathcal{A}_i$, and receives a reward $r_{i,t}$. Agent i's decision, given the system state $s_t \in \mathcal{S}$, is determined by the policy $\pi_{\theta_i}(\cdot|s_t)$. This policy is a density function mapping from $\mathcal{A}_i$ to $[0, \infty)$ and is parameterized by θ_i. Here, θ_i is an element of a generic set $\Theta_i \in \mathcal{R}^{m_i}$, where the dimensionality m_i is determined by each agent. In choosing m_i, each agent faces a tradeoff: it should be large enough to avoid underfitting, yet not too large to risk overfitting and increased training time. Let $\theta := [\theta_1, \cdots, \theta_I]$ and $\pi_\theta(\cdot|s) = \prod_{i=1}^{I} \pi_{\theta_i}(\cdot|s)$ be the joint policy of all agents. In the following discussion of this section, we assume that the Markov chains $\{s_t\}_{t\ge 0}$ induced by the collective policy π_θ is ergodic with a stationary distribution ρ_θ.

[3] Sufficient conditions under which (7) has a solution are provided in [23].

Different from a fully decentralized framework, the goal for each agent in the consensus-update framework is to solve the following optimization problem:

$$\sup_{\theta \in \Theta := \Pi_{i=1}^{I} \Theta_i} J(\theta) = \mathbb{E}_{s_0 \sim \rho_\theta}\left[\lim_{T\to\infty} \frac{1}{T} \sum_{t=0}^{T} \left(\frac{1}{I} \sum_{i=1}^{I} r_{i,t}\right) \middle| s_0, \pi_\theta \right], \tag{9}$$

which is to optimize the average system long-term reward. The objective function resembles a cooperative game, which may face similar critiques regarding the need for each agent to access system-wide state variables in a decentralized setting. A pertinent response, as previously mentioned, is the use of control automation: the objective function can be integrated into intelligent control devices, making them 'black boxes' for end-users. Additionally, in developing a MARL framework with provable convergence, some level of global information sharing among agents is likely unaoidable. The collective objective function, as outlined in (9), is essential for decentralized agents to reach a consensus.

To ensure that the objective function in (9) is well-defined, we have the following result.

Lemma IV.1. *The supreumum in* (9)*,* $\sup_\theta J(\theta)$*, exists and is finite.*

Proof. Since the market clearing price P_t, the supply-demand ratio SDR_t, and agents bid $b_{i,t}$ for $i = 1, \ldots, I$ are all bounded above, the first term in an agent's reward function, R_i^m, is then bounded above based on Eq. (5), for any $a \in \mathcal{A} := \Pi_{i=1}^{I} \mathcal{A}_i$ and $s \in \mathcal{S}$. The second term, R^v, is bounded above by 0 based on its definition in Eq. (8). Hence, agent i's reward $r_{i,t}$ is bounded at any t, and for any $a \in \mathcal{A}$ and $s \in \mathcal{S}$. Since the number of agents is finite, there exists a common upper bound for all $r_{i,t}$'s, and let us denote it as $\overline{R}$. Then by the formulation in (9), $J(\theta) \le \overline{R}$ for all $\theta \in \Theta$. By the well-known least-upper-bound property of real numbers, we know that $\sup_\theta J(\theta)$ exists and is finite. $\square$

B. Consensus-update Actor-critic Algorithm

In this section, we introduce the consensus-update actor-critic algorithm for MARL with continuous state and action spaces, developed in [1], and apply it to solve Problem (9). First, define the global relative action-value function $Q_\theta(s, a)$ for a given $a \in \mathcal{A}$, $s \in \mathcal{S}$ and a policy π_θ as:

$$Q_\theta(s, a) := \sum_{t=0}^{\infty} \mathbb{E}\left[\frac{1}{I} \sum_{i=1}^{I} r_{i,t} - J(\theta) \mid s_0 = s, a_0 = a, \pi_\theta \right]. \tag{10}$$

Note that since all the agents share the same $J(\theta)$, the Q function does not need to have an agent index and is the same across the agents.

The function $Q_\theta(s, a)$ cannot be calculated by each agent, even if the global state and action information are shared, since the joint policy distribution π_θ is not known. Instead, each agent uses $\hat{Q}(\cdot, \cdot; \omega^i) : \mathcal{S} \times \mathcal{A} \to \mathcal{R}$, which is a class of functions parametrized by $\omega^i \in \mathcal{R}^K$, where K is the dimension of the ω^i, to approximate the action-value function $Q_\theta(s, a)$. Note that unlike the parameters θ_i, where agents can choose their own policy models (and hence different dimensions of θ_i), the dimension of ω^i has to be the same across all agents (and hence K does not have an agent index) to facilitate the consensus update.

The decentralized consensus-update actor-critic algorithm for networked multi-agents works as follows: at the beginning of time step $t+1$, each agent i observes the global state s_{t+1} and chooses their action $a_{i,t+1}$ according to their own policy $\pi_{\theta_{i,t}}$ (referred to as the 'actor'), where $\theta_{i,t}$ is the policy parameter for agent i at time step t. Then each agent i will receive the joint action $a_{t+1} = (a_{1,t+1}, \ldots, a_{I,t+1})$ and their own reward $r_{i,t+1}$, which help them update $\hat{Q}(\cdot,\cdot;\omega_t^i)$ (referred to as the 'critic') and $\pi_{\theta_{i,t}}$ on their own in each time step. Specifically, each agent first updates the temporal-difference (TD) error δ_t^i and long-term return estimate $\bar{r}_{t+1}^i$ as follows:

$$\delta_t^i := r_{i,t+1} - \bar{r}_{t+1}^i + \hat{Q}\left(s_{t+1}, a_{t+1}; \omega_t^i\right) - \hat{Q}\left(s_t, a_t; \omega_t^i\right) \tag{11}$$

$$\bar{r}_{t+1}^i \leftarrow (1-\beta_{\omega,t}) \cdot \bar{r}_t^i + \beta_{\omega,t} \cdot r_{i,t} \tag{12}$$

Let $\widetilde{\omega}_t^i$ denote a temporary local parameter of $\hat{Q}(\cdot,\cdot;\omega_t^i)$ for agent i, and it is updated as:

$$\widetilde{\omega}_t^i = \omega_t^i + \beta_{\omega,t} \cdot \delta_t^i \cdot \nabla_\omega \hat{Q}\left(s_t, a_t; \omega_t^i\right). \tag{13}$$

The local parameter of the policy $\pi_{\theta_{i,t}}$ is updated as

$$\theta_{i,t+1} = \Gamma^i[\theta_{i,t} + \beta_{\theta,t} \cdot \widehat{I}_t^{Q,i}], \tag{14}$$

where $\Gamma^i : \mathbb{R}^{m_i} \rightarrow \Theta_i \subset \mathbb{R}^{m_i}$ is a projector operator. An example is the orthogonal projector mapping: $\Gamma^i(x) \triangleq \arg\min_{y\in\Theta_i} \|y-x\|$, $\forall \mathbf{x} \in \mathbb{R}^{m_i}$, where Θ is a compact convex set. The $\widehat{I}_t^{Q,i}$ term in (14) is given as follows:

$$\widehat{I}_t^{Q,i} = \int_{\mathcal{A}^i} d\pi_{\theta_{i,t}}\left(a^i \mid s_t\right) \nabla_{\theta_i} \log \pi_{\theta_{i,t}}\left(a^i \mid s_t\right) \hat{Q}\left(s_t, a^i, a_t^{-i}; \omega_t^i\right). \tag{15}$$

To limit communication among agents, each agent i only shares local parameter $\widetilde{\omega}_t^i$ with each other; no other information needs to be shared. A consensual estimate of $Q_\theta(s,a)$ can then be estimated through updating the parameter ω_{t+1}^i as follows:

$$\omega_{t+1}^i = \sum_{j=1}^{I} \widetilde{\omega}_t^j. \tag{16}$$

To apply such an algorithm to P2P energy trading, we illustrate the conceptual framework in Fig. 2. The detailed algorithm is presented in Algorithm 1.

Algorithm 1: Consensus-update actor-critic algorithm for voltage control with P2P energy market

Input: Initial values of the parameters $\bar{r}_0^i, \omega_0^i, \widetilde{\omega}_0^i, \theta_0^i, \forall i \in \{1,\ldots,I\}$, the initial state s_0, and stepsizes $\{\beta_{\omega,t}\}_{t\geq 0}$ and $\{\beta_{\theta,t}\}_{t\geq 0}$.

for each agent $i = 1, \ldots, I$ **do**
  Makes the decision on reactive power generation and battery charging/discharging $a_{i,0} \sim \pi_{\theta_0^i}(\cdot \mid s_0)$.
end
The power flow equation (7) is solved, the P2P energy market is cleared, and the information of joint actions $a_0 = (a_{i,0}, \ldots, a_{I,0})$ is sent to each agent.
for each agent $i = 1, \ldots, I$ **do**
  Observes the reward $r_{i,0}$.
end
for each time step $t = 0, 1, \ldots T$ **do**
  for each agent $i = 1, \ldots, I$ **do**
    Observes the next global state s_{t+1}.
    Update $\bar{r}_{t+1}^i \leftarrow (1-\beta_{\omega,t}) \cdot \bar{r}_t^i + \beta_{\omega,t} \cdot r_{i,t}$.
    Makes the decision on reactive power generation and battery charging/discharging $a_{i,t+1} \sim \pi_{\theta_{i,t}}(\cdot \mid s_{t+1})$.
  end
  The power flow equation (7) is solved, the P2P energy market is cleared, and the information of joint actions $a_{t+1} = (a_{i,t+1}, \ldots, a_{I,t+1})$ is sent to each agent.
  for each agent $i = 1, \ldots, I$ **do**
    1. Observes the reward $r_{i,t+1}$ and update $\delta_t^i \leftarrow r_{i,t+1} - \bar{r}_{t+1}^i + \hat{Q}\left(s_{t+1}, a_{t+1}; \omega_t^i\right) - \hat{Q}\left(s_t, a_t; \omega_t^i\right)$.
    2. **Critic step:** $\widetilde{\omega}_t^i \leftarrow \omega_t^i + \beta_{\omega,t} \cdot \delta_t^i \cdot \nabla_\omega \hat{Q}\left(s_t, a_t; \omega_t^i\right)$.
    3. **Actor step:** $\theta_{i,t+1} \leftarrow \Gamma^i[\theta_{i,t} + \beta_{\theta,t} \cdot \widehat{I}_t^{Q,i}]$, where $\Gamma^i : \mathbb{R}^{m_i} \rightarrow \Theta_i \subset \mathbb{R}^{m_i}$, and

$$\widehat{I}_t^{Q,i} = \int_{\mathcal{A}^i} d\pi_{\theta_{i,t}}\left(a^i \mid s_t\right) \cdot \tag{17}$$

$$\nabla_{\theta_i} \log \pi_{\theta_{i,t}}\left(a^i \mid s_t\right) \hat{Q}\left(s_t, a^i, a_t^{-i}; \omega_t^i\right), \tag{18}$$

    4. Send $\widetilde{\omega}_t^i$ to the other agent in the system.
  end
  for each agent $i = 1, \ldots, I$ **do**
    Consensus step: $\omega_{t+1}^i \leftarrow \sum_{j=1}^{I} \widetilde{\omega}_t^j$.
  end
end

C. Convergence Results

In this section, we discuss the convergence results of Algorithm 1. For ease of notation, we drop the t index of the parameters θ_i for $i \in \mathcal{I}$ in the following discussions. First, we need a technical condition of Markov chains, known as the geometric ergodicity.

Definition IV.1. *(Geometric ergodicity, referred to as uniform ergodicity in [24]) A Markov chain having stationary distribution $\pi(\cdot)$ is geometricly ergodic if*

$$||P^m(x,\cdot) - \pi(\cdot)|| \leq Mc^m, \; m = 1, 2, 3, \ldots$$

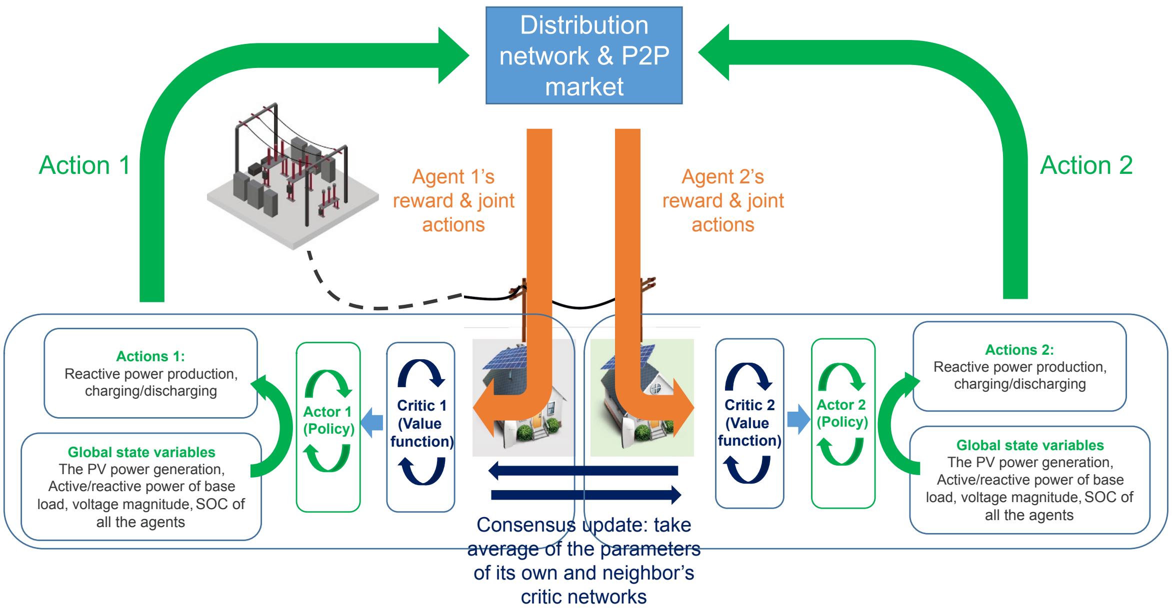

Figure 2: Consensus MARL Framework for the Voltage Control with P2P Market

for some $0 < c < 1$ *and* $0 < M < \infty$*, where* P *is the transition kernel of the Markov chain.*

The concept of geometric ergodicity specifies the rate at which a Markov chain converges to a stationary distribution if it exists. In the following, we state the first assumption to ensure the consensus MARL convergence to a steady state, which requires the Markov chains formed by the system state and state-action pairs to be both geometrically ergodic. Granted that this assumption cannot be easily verified, it is essential to the convergence proof.

Assumption 1. *For each* $i \in \mathcal{I}, s \in \mathcal{S}$ *and* $\theta_i \in \Theta_i$*, agent* i*'s policy is positive; that is,* $\pi_{\theta_i}(a_i \mid s) > 0$ *for all* $a_i \in \mathcal{A}_i$*. Additionally,* $\pi_{\theta_i}(\cdot \mid s)$ *is assumed to be continuously differentiable with respect to the parameter* θ_i *over* Θ_i*. Finally, the Markov chains* $\{s_t\}_{t\geq 0}$ *and* $\{(s_t, a_t)\}_{t\geq 0}$ *induced by the agents' collective policies* π_θ *are both geometrically ergodic, with the coressponding stationary distribution denoted by* ρ_θ *and* $\tilde{\rho}_\theta$*, respectively.*

Remark 1 The first part of Assumption 1 is a standard assumption on policy functions. One example of such a policy is Gaussian: $\pi_{\theta_i}(\cdot \mid s) \sim \mathcal{N}(\eta_{\theta_i}(s), \Sigma_i)$, where $\eta_{\theta_i}(s) : \mathcal{S} \to \mathcal{A}_i \in \mathbb{R}^n$ is a fully connected neural network parametrized by θ_i and is continuously differentiable with respect to θ_i and s, and $\Sigma_i \in \mathbb{R}^{n\times n}$ is the covariance matrix.

Assumption 2. *The policy parameter* θ_i *for each agent is updated by a local projection operator,* $\Gamma^i : \mathbb{R}^{m_i} \to \Theta_i \subset \mathbb{R}^{m_i}$*, that projects any* θ_i *onto a compact set* Θ_i*. In addition,* $\Theta = \prod_{i=1}^{I} \Theta_i$ *includes at least one local minimum of* $J(\theta)$*.*

Remark 2 The requirement of the local projection operator is standard in convergence analyses of many reinforcement learning algorithms. If Θ_i is a convex set, a qualifying example of Γ^i can be the nearest point projection on Θ_i; that is,

$$\Gamma^i(\theta_i) = \arg\max_{\theta_i{}^* \in \Theta_i} \|\theta_i - \theta_i{}^*\|_2. \tag{19}$$

Next, we make an assumption on the action-value function approximation.

Assumption 3. *For each agent* i*, the action-value function is approximated by linear functions, i.e.,* $\hat{Q}(s, a; \omega) = \omega^\top \phi(s, a)$ *where* $\phi(s, a) = [\phi_1(s, a), \cdots, \phi_K(s, a)]^\top \in \mathbb{R}^K$ *is the feature associated with* (s, a)*. The feature function,* $\phi_k : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$*, for* $k = 1, \cdots, K$*, is bounded for any* $s \in \mathcal{S}, a \in \mathcal{A}$*. Furthermore, the feature functions* $\{\phi_k\}_{k=1}^K$ *are linearly independent, and for any* $u \in \mathbb{R}^K$ *and* $u \neq 0, u^\top \phi$ *is not a constant function over* $\mathcal{S} \times \mathcal{A}$*.*

Remark 3. One exmaple of the action-value function approximation is the Gaussian radial basis function (RBF):

$$\hat{Q}(s, a; \omega) = \sum_{j=1}^{K} \omega_i e^{-\gamma_j \|[s,a]-c_j\|}, \tag{20}$$

where $[s, a]$ is the concatenation of s and a, $\gamma_j \in \mathcal{R}^+$ for $j = 1, \cdots, K$, $c_j \in \mathcal{R}^{|\mathcal{S}|+|\mathcal{A}|}$ for $j = 1, \cdots, K$. The parameters γ_j and c_j can be chosen arbitrarily, as long as (γ_j, c_j) are different with different j so that the feature functions are linearly independent.

Assumption 4. *The stepsizes* $\beta_{\omega,t}$ *and* $\beta_{\theta,t}$ *in Algorithm 1 satisfy*

$$\sum_t \beta_{\omega,t} = \sum_t \beta_{\theta,t} = \infty, \text{ and } \sum_t (\beta_{\omega,t}^2 + \beta_{\theta,t}^2) < \infty. \tag{21}$$

Also, $\beta_{\theta,t} = o(\beta_{\omega,t})$ *and* $\lim_{t\to\infty} \beta_{\omega,t+1} \cdot \beta_{\omega,t}^{-1} = 1$*.*

An example of such stepsizes can be $\beta_{\omega,t} = 1/t^{0.8}$ and $\beta_{\theta,t} = 1/t$.

With the above assumptions, the convergence of Algorithm 1 is given below.

Theorem IV.1. *Assume that each agent's state space $\mathcal{S}_i$ is compact for $i \in \mathcal{I}$. Under Assumptions 1 – 4, the sequences $\{\theta_{i,t}\}$, $\{\mu_t^i\}$ and $\{\omega_t^i\}$ generated from Algorithm 1 satisfy the following.*

(i) *Convergence of the critic step: $\frac{1}{I}\lim_{t\to\infty}\sum_{i\in\mathcal{I}}\mu_t^i = J(\theta)$, where $J(\theta)$ is defined in* (9)*, and $\lim_{t\to\infty}\omega_t^i = \omega_\theta$ for all $i \in \mathcal{I}$, with ω_θ being a solution of a fixed point mapping. (The convergence is in the sense of almost surely.)*

(ii) *Convergence of the actor step: for each $i \in \mathcal{I}$, $\theta_{i,t}$ converges almost surely to a point $\hat{\theta}_i$ that is a projection onto the set Θ.*

Proof. The general convergence result of the consensus algorithm is proved in [1] under six assumptions. Our Assumption 1, 2, 3 and 4 directly correspond to four assumptions in [1]. The fifth assumption in [1] regarding agents' time-varying neighborhood is trivially true here since we assume agents can communicate through the whole network, and the network topology does not change over time. Hence, the only remaining item to show is the uniform boundedness of agents' reward $r_{i,t}$. By the definition of the market clearing price P_t in (2), it is bounded between FIT and UR. Together with the compactness assumption of the state space $\mathcal{S}_i$ (which is reasonable since an agent's energy demand and battery/PV capacities are all bounded), the R_i^m component in the reward function, as defined in (4), is uniformly bounded. The other term, the voltage violation penalty R_i^v, is uniformly upper bounded by 0 and lower bounded by $-\lambda NM$. Hence, $r_{i,t}$ is uniformly bounded for all i and t, and the convergence results follow directly from Theorem 2 and 3 in [1]. □

Remark 4. While the theorem establishes the convergence to a steady state of the critic and actor steps, it does not address how good the system reward is upon convergence. A key concern in this context is the potential inadequacy of the linear approximation of the critic function $\hat{Q}(s, a; \omega)$ in Assumption 3. In our model, the distribution network encompasses a multitude of agents, each exhibiting non-linear behaviors and interactions that may not be adequately captured by simple linear models. One natural idea is to deep neural networks (DNNs) instead, as they excel in capturing and modeling non-linear relationships. The downside of using DNNs, however, is that the proof of Theorem IV.1 is no longer applicable. Consequently, the convergence of the critic and actor steps remains an unresolved issue, meriting further investigation in future research.

V. Simulation Results

A. Test Data

We test the algorithm frameworks using the IEEE 13-bus test feeder [25]. In our simulations, we set each bus to have one prosumer, excluding the substation bus; hence, there are 12 prosumers (and no pure consumers). The length of each time step is set to be 1 hour. The distribution network's voltage limit is set to $[0.96\,pu, 1.04\,pu]$. The amplifying parameter for voltage violation, λ, is set at 10^4. The price ceiling and floor, that is, the utility rate UR and feed-in tariff FIT, are set at 14 ¢/kWh and 5 ¢/kWh, respectively. The capacity of each agent's PV is set to be 30kW, and the energy storage capacity is set to be 50kWh, with the charging and discharging efficiency being 0.95 and 0.9 for each agent. The PV generation of the prosumers and the total base load of the pure consumers on each bus have fixed diurnal shapes. These shapes represent the average values corresponding to each hour of the day. For a detailed explanation and due to page limitations her, please refer to our earlier paper [2]. Additionally, [2] provides more information on how the random PV output for each agent is generated. The power flow equations (7) are solved by the open-source distribution system simulator OpenDSS [26].

B. Deep Neural Network Implementation

As stated in Remark 4, we use DNNs instead of linear combinations of features to approximate the critic function $\hat{Q}$. To implement the consensus MARL with DNNs, we apply the default architectures of the actor and critic networks of actor-critic algorithms in RLlib [27]. To be specific, for the policy function, we use

$$\pi_{\theta_i}(\cdot|s) \sim \mathcal{N}\left(\begin{bmatrix}\sigma_{\theta_i}^{(1)}(s)\\ \sigma_{\theta_i}^{(2)}(s)\end{bmatrix}, \begin{bmatrix}\sigma_{\theta_i}^{(3)}(s) & 0\\ 0 & \sigma_{\theta_i}^{(4)}(s)\end{bmatrix}\right), \quad (22)$$

where $\sigma_{\theta_i}^{(1)}(s)$ represents the mean value of the charging/discharging action, and $\sigma_{\theta_i}^{(2)}(s)$ denotes the mean value of the smart inverter action. Similarly, $\sigma_{\theta_i}^{(3)}(s)$ and $\sigma_{\theta_i}^{(4)}(s)$ correspond to the variances of the charging/discharging action and the smart inverter action, respectively. The function $\sigma_{\theta_i}(s) : \mathcal{S} \to \mathcal{R}^4$ is implemented as a DNN, parameterized by θ_i. This network comprises two fully connected layers, each consisting of 256 neurons, and employs a $tanh$ activate function following each layer. A similar neural network architecture is also applied to the approximator of the action-value function $\hat{Q}(\cdot, \cdot; \omega^i)$. The settings of stepsizes are $\beta_{\omega,t} = 1/t^{0.65}$ and $\beta_{\theta,t} = 1/t^{0.85}$.

C. Comaprison of Three Different MARL Algorithms

We compare the consensus MARL with two other algorithm frameworks: (i) a fully decentralized framework in which each agent just solves their own reinforcement learning problem using the PPO algorithm [2], while completely ignoring multiagent interactions, and (ii) the MADDPG approach [4].

Different from maximizing the system's total long-term expected reward, the objective for each agent in the fully decentralized framework and in MADDPG is to optimize their own long-term expected reward:

$$\sup_{\theta_i} J_i(\theta_i) = \mathbb{E}_{s_0\sim\rho_\theta}\left[\sum_{t=0}^{\infty}\gamma_i^t r_{i,t}|s_0, \pi_\theta\right], \; i = 1, \ldots, I, \quad (23)$$

where γ_i is the discount factor of agent i. In the fully decentralized framework, each agent uses the single-agent PPO algorithm from [3] to update their own policy and value function locally based on only their own state, action, and reward information.

Since all three MARL approaches are categorized as policy gradient descent methods, they update the parameter θ_{t+1} in a similar manner to approximate gradient ascent of $J(\theta)$ as follows:

$$\theta_{t+1} = \theta_t + \beta\widehat{\nabla J(\theta_t)}, \quad (24)$$

where β is the step size and $\widehat{\nabla J(\theta_t)}$ is a stochastic estimate approximating the gradient of the performance measure. The three MARL approaches differ in how each agent updates $\widehat{\nabla J(\theta_t)}$. To highlight such differences, we present the specific gradient approximations for the three approaches below.

In the fully decentralized approach, the gradient update uses local states and actions only: for each $i \in \mathcal{I}$,

$$\nabla_{\theta_i} \widehat{J_i(\theta_i)} = \mathbb{E}_{s\sim\rho^\theta, a_i\sim\pi_{\theta_i}} \left[\nabla_{\theta_i} \log \pi_{\theta_i}(a_i \mid s_i)\, Q_i^\pi(s_i; a_i)\right]. \tag{25}$$

In the MADDPG approach, each agent makes decisions based on only their own state variables. However, their action-value functions are updated based on the global states and actions of all agents by a central authority, who is assumed to have access to the global information. Specifically, for each agent $i \in \mathcal{I}$, the gradient approximation in MADDPG is

$$\begin{aligned} &\nabla_{\theta_i} \widehat{J_i(\theta_i)} = \\ &\quad \mathbb{E}_{s\sim\rho^\theta, a_i\sim\pi_{\theta_i}} \left[\nabla_{\theta_i} \log \pi_{\theta_i}(a_i \mid s_i)\, Q_i^\pi(\boldsymbol{s}; \boldsymbol{a_1}, \ldots, \boldsymbol{a_I})\right], \end{aligned} \tag{26}$$

where $Q_i^\pi(\boldsymbol{s}; \boldsymbol{a_1}, \ldots, \boldsymbol{a_I})$ is the centralized action-value function to be updated by the assumed central authority. The policy function, $\pi_{\theta_i}(a_i \mid s_i)$, on the other hand, uses only local states s_i. This is what is referred to as centralized training and decentralized execution.

For the consensus framework, the gradient estimation uses the so-called expected policy gradient as in [1]:

$$\nabla_{\theta_i} \widehat{J(\theta)} = \mathbb{E}_{s\sim\rho^\theta, a_{-i}\sim\pi_{\theta_{-i}}} \mathcal{I}^Q_{\theta_i}(s, a_{-i}), \; i = 1, \ldots, I, \tag{27}$$

where

$$\mathcal{I}^Q_{\theta_i}(s, a_{-i}) = \mathbb{E}_{a_i\sim\pi_{\theta_i}} \nabla_{\theta_i} \log \pi_{\theta_i}(a_i \mid \boldsymbol{s})\, Q_i^\pi(\boldsymbol{s}; \boldsymbol{a_1}, \ldots, \boldsymbol{a_I}). \tag{28}$$

While the action-value function Q_i^π is the same as in (26), the key difference lies in the policy function. As stated in Section IV-A and reflected in (28), each agent i's policy depends on all agents' states $\boldsymbol{s}$. The other difference between the consensus MARL and the MADDPG algorithm is how the action-value function Q_i^π is updated. In MADDPG, the function is updated through centralized training by a central authority; while in consensus MARL, updates are executed via a decentralized consensus process, as described in Section IV-B.

D. Numerical Results

Figure 3 displays the convergence curves of the 30-episode moving average of the episodic total reward in the distribution network for the three different MARL algorithms. It is evident that the mean episodic total reward gradually converges to a high level at the end of the training in all three frameworks. The results show that the consensus MARL algorithm outperforms the fully decentralized framework, which is in line with expectations, since the consensus-update framework benefits from improved approximation of the action-value function through communication. The better performance of consensus MARL over MADDPG, on the other hand, is surprising. Conventionally, one might anticipate a centralized algorithm to outperform decentralized algorithms. The explanation here can be traced to MADDPG's decentralized execution. In MADDPG, each agent independently updates its actions based on its own local state, despite the centralized nature of the learning process. This approach of relying solely on local information appears to be less effective compared to the consensus update mechanism.

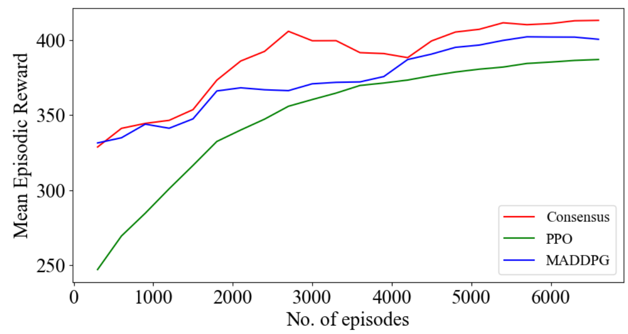

Figure 3: 30-episode moving average of episodic total reward

Figure 4 presents the market prices of the P2P market in chronological order over the last three days after thousands of training episodes. Notably, the market price in the consensus-update framework demonstrates a tendency to be higher during off-peak hours and lower during peak hours. This indicates that agents employing the consensus-update algorithm have developed more sophisticated trading strategies than the other two MARL frameworks. Specifically, the consensus-algorithm-induced strategies involve buying more and selling less during off-peak hours, and conversely, selling more and buying less during peak hours, when compared with the strategies produced by the purely decentralized algorithm.

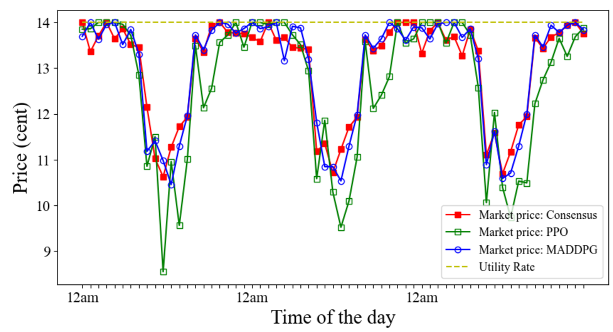

Figure 4: Hourly prices over the last three days

Figure 5 shows the gradual reduction of the system voltage deviation after training. Initially, it is observed that voltage violations are significantly high across all three frameworks. However, the consensus framework demonstrates a rapid decline in these violations, converging swiftly and effectively to zero at the fastest rate. The remaining two frameworks demonstrate a slower rate of convergence, with the fully decentralized approach seemingly unable to fully eliminate voltage violations.

VI. Conclusion and Future Research

This study has developed a market and algorithmic framework to enable consumers' and prosumers' participation in local P2P energy trading. Utilizing reinforcement learning algorithms, we have automated bidding for agents while ensuring de-

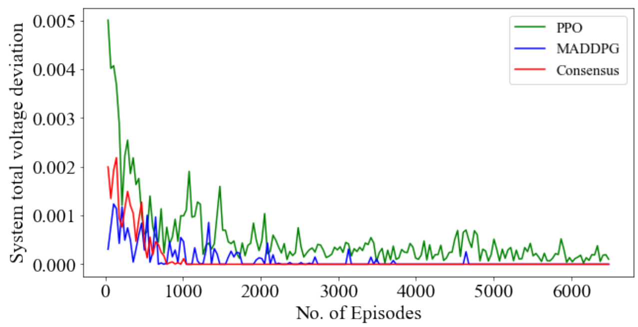

Figure 5: System voltage deviation. The total system voltage deviation (p.u.) in the n-th episode is calculated as $\sum_{t=24(n-1)}^{24n-1} \sum_{j:Bus}[\max(0, v_t^j - \bar{v}) + \max(0, \underline{v} - v_t^j)]$.

centralized decision-making. The SDR-based market clearing addresses challenges with zero-marginal-cost resources and simplifies bidding. Additionally, our MARL framework includes voltage constraints of the physical network, setting a foundation for real-world applications.

However, this research represents just the initial phase in the practical application of a P2P energy trading market, with several challenges ahead. Theoretically, the scalability of the MARL framework is crucial; if the scalability of the consensus MARL algorithm is limited, a mean-field approach, as in [28], could be considered, where agents operate with the belief that the market outcome is at a mean-field equilibrium. Practically, while the current model uses discrete, synchronous trading rounds, future work should investigate continuous and asynchronous trading among agents.

Additionally, while integrating transmission constraints into the current framework is straightforward (that is, they can be added to the power flow equations (7)), the allocation of transmission losses raises important equity considerations that must be addressed.

Another critical aspect is cybersecurity. With increased automation, the system becomes vulnerable to cyber attacks, such as malicious users compromising smart inverters to inject false information into the market. In this context, the work in [21] provides valuable insights.

Last but not least, it is essential to investigate how short-term P2P market dynamics influence long-term consumer investment decisions, particularly regarding the adoption of solar panels and energy storage systems.

References

[1] K. Zhang, Z. Yang, and T. Basar, "Networked multi-agent reinforcement learning in continuous spaces," in 2018 IEEE conference on decision and control (CDC), pp. 2771–2776, IEEE, 2018.

[2] C. Feng, A. L. Lu, and Y. Chen, "Decentralized voltage control with peer-to-peer energy trading in a distribution network," in Proceedings of the 56th Hawaii International Conference on System Sciences, pp. 2600 – 2609, 2023.

[3] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, "Proximal policy optimization algorithms," arXiv preprint arXiv:1707.06347, 2017.

[4] R. Lowe, Y. I. Wu, A. Tamar, J. Harb, O. Pieter Abbeel, and I. Mordatch, "Multi-agent actor-critic for mixed cooperative-competitive environments," Advances in neural information processing systems, vol. 30, 2017.

[5] N. Liu, X. Yu, C. Wang, C. Li, L. Ma, and J. Lei, "Energy-sharing model with price-based demand response for microgrids of peer-to-peer prosumers," IEEE Transactions on Power Systems, vol. 32, no. 5, pp. 3569–3583, 2017.

[6] T. Morstyn and M. D. McCulloch, "Multiclass energy management for peer-to-peer energy trading driven by prosumer preferences," IEEE Transactions on Power Systems, vol. 34, no. 5, pp. 4005–4014, 2018.

[7] M. H. Ullah and J.-D. Park, "Peer-to-peer energy trading in transactive markets considering physical network constraints," IEEE Transactions on Smart Grid, vol. 12, no. 4, pp. 3390–3403, 2021.

[8] Y. Liu, C. Sun, A. Paudel, Y. Gao, Y. Li, H. B. Gooi, and J. Zhu, "Fully decentralized P2P energy trading in active distribution networks with voltage regulation," IEEE Transactions on Smart Grid, vol. 14, no. 2, pp. 1466–1481, 2022.

[9] Y. Zhang, E. Dall'Anese, and M. Hong, "Online proximal-admm for time-varying constrained convex optimization," IEEE Transactions on Signal and Information Processing over Networks, vol. 7, pp. 144–155, 2021.

[10] W. Tushar, T. K. Saha, C. Yuen, D. Smith, and H. V. Poor, "Peer-to-peer trading in electricity networks: An overview," IEEE Transactions on Smart Grid, vol. 11, no. 4, pp. 3185 – 3200, 2020.

[11] W. Tushar, C. Yuen, T. K. Saha, T. Morstyn, A. C. Chapman, M. J. E. Alam, S. Hanif, and H. V. Poor, "Peer-to-peer energy systems for connected communities: A review of recent advances and emerging challenges," Applied energy, vol. 282, p. 116131, 2021.

[12] W. Tushar, C. Yuen, H. Mohsenian-Rad, T. Saha, H. V. Poor, and K. L. Wood, "Transforming energy networks via peer-to-peer energy trading: The potential of game-theoretic approaches," IEEE Signal Processing Magazine, vol. 35, no. 4, pp. 90–111, 2018.

[13] E. A. Soto, L. B. Bosman, E. Wollega, and W. D. Leon-Salas, "Peer-to-peer energy trading: A review of the literature," Applied Energy, vol. 283, p. 116268, 2021.

[14] J. Guerrero, A. C. Chapman, and G. Verbič, "Decentralized P2P energy trading under network constraints in a low-voltage network," IEEE Transactions on Smart Grid, vol. 10, no. 5, pp. 5163–5173, 2018.

[15] D. Biagioni, X. Zhang, D. Wald, D. Vaidhynathan, R. Chintala, J. King, and A. S. Zamzam, "Powergridworld: A framework for multi-agent reinforcement learning in power systems," arXiv preprint arXiv:2111.05969, 2021.

[16] S. Wang, J. Duan, D. Shi, C. Xu, H. Li, R. Diao, and Z. Wang, "A data-driven multi-agent autonomous voltage control framework using deep reinforcement learning," IEEE Transactions on Power Systems, vol. 35, no. 6, pp. 4644–4654, 2020.

[17] D. Qiu, J. Wang, J. Wang, and G. Strbac, "Multi-agent reinforcement learning for automated peer-to-peer energy trading in double-side auction market.," in IJCAI, pp. 2913–2920, 2021.

[18] K. Zhang, Z. Yang, H. Liu, T. Zhang, and T. Basar, "Fully decentralized multi-agent reinforcement learning with networked agents," in International Conference on Machine Learning, pp. 5872–5881, PMLR, 2018.

[19] G. Qu, A. Wierman, and N. Li, "Scalable reinforcement learning for multiagent networked systems," Operations Research, vol. 70, no. 6, pp. 3601–3628, 2022.

[20] Z. Zhao, C. Feng, and A. L. Liu, "Comparisons of auction designs through multi-agent learning in peer-to-peer energy trading," IEEE Transactions on Smart Grid, 2022. Online first.

[21] M. Figura, K. C. Kosaraju, and V. Gupta, "Adversarial attacks in consensus-based multi-agent reinforcement learning," in 2021 American Control Conference (ACC), pp. 3050–3055, IEEE, 2021.

[22] P. Huang, A. Scheller-Wolf, and K. Sycara, "Design of a multi–unit double auction e–market," Computational Intelligence, vol. 18, no. 4, pp. 596–617, 2002.

[23] B. Cui and X. A. Sun, "Solvability of power flow equations through existence and uniqueness of complex fixed point." arXiv preprint arXiv:1904.08855, 2019.

[24] G. O. Roberts and J. S. Rosenthal, "General state space markov chains and MCMC algorithms," Probability Surveys, vol. 1, pp. 20–71, 2004.

[25] IEEE Distribution System Analysis Subcommittee, "IEEE 13 Node Test Feeder." https://cmte.ieee.org/pes-testfeeders/resources/.

[26] Electric Power Research Institute, "OpenDSS." https://www.epri.com/pages/sa/opendss.

[27] The Ray Team, "Models, preprocessors, and action distributions Ray 2.8.1." https://docs.ray.io/en/latest/rllib/rllib-models.html#default-model-config-settings.

[28] X. Guo, A. Hu, R. Xu, and J. Zhang, "Learning mean-field games," Advances in neural information processing systems, vol. 32, 2019.