Multi-Agent Deep Hedging: Benchmarking Prosumer Strategies on Electricity Trading Platforms

Nicolas Eschenbaum, Nicolas Greber and Oleg Szehr*

December 15, 2025

Abstract

We introduce Multi-Agent Deep Hedging (MADH), a computational framework that extends deep reinforcement learning to markets with endogenous price formation. MADH embeds a differentiable market-clearing mechanism into the learning process, enabling decentralized agents to internalize their price impact via gradient ascent. We apply MADH to peer-to-peer electricity trading, benchmarking it against a centralized welfare-maximizing planner. Using synthetic data for heterogeneous prosumer communities, we demonstrate that decentralized agents autonomously learn sophisticated arbitrage strategies, such as capacity withholding. Crucially, we find that this strategic behavior generates positive externalities: while active traders reduce their own costs through price-awareness, their arbitrage smooths market prices, reducing costs for passive consumers. Furthermore, quantitative regret analysis confirms that MADH policies converge with low regret $< 1.5\%$. These results establish MADH as a scalable tool for designing stable and efficient autonomous trading platforms.

*Eschenbaum: Swiss Economics, Ottikerstr. 7, 8006 Zurich, Switzerland (nicolas.eschenbaum@swiss-economics.ch); Greber: Swiss Economics, Ottikerstr. 7, 8006 Zurich, Switzerland (nicolas.greber@swiss-economics.ch); Szehr: Dalle Molle Institute for Artificial Intelligence, Via la Santa 1, 6962 Viganello, Switzerland (oleg.szehr@idsia.ch). We gratefully acknowledge financial support from the Swiss Federal Office of Energy through grant no. SI/502525

1 Introduction

A common assumption in automated pricing and risk management is that individual trades have negligible influence on market prices. In markets with few active participants or large agents, this small-investor assumption is inappropriate. We introduce *Multi-Agent Deep Hedging* (MADH), which extends Deep Hedging to a multi-agent reinforcement learning (MARL) setting and formulates the market as a Markov game with endogenous prices. We apply MADH to peer-to-peer (P2P) electricity trading among prosumers, using it to study price formation and the battery-charging and curtailment policies that emerge, as well as scalability and welfare implications.

1.1 Multi-Agent Deep Hedging

Deep Hedging (DH) (Buehler et al., 2019) models multi-period decision problems with a stack of shallow policies—one per time step—with gradients propagated end-to-end to optimize a terminal objective. In its standard form, DH treats market prices as exogenous and does not include a market-clearing mechanism. This approach has found broad applications in traditional finance (Krabichler and Teichmann, 2023) and strategic planning in energy consumption and storage (Curin et al., 2021).

We extend DH to a multi-agent setting. Each agent is equipped with a neural-network policy, and the market-clearing price is modeled as a differentiable function of all agents' actions. This allows each agent to internalize price impact via joint gradient updates through the full computation graph. Conceptually, MADH phrases the environment as a Markov game and supports both a decentralized training and decentralized execution (DTDE) setup and a centralized training and centralized execution (CTCE) benchmark.

Because deterministic policy methods can under-explore—especially in stochastic, non-stationary MARL environments (e.g., Mnih et al. (2015); Silver et al. (2016); Lillicrap et al. (2015))—we complement passive exploration from environmental randomness with explicit policy stochasticity ("active noise") during training. This improves robustness and helps avoid convergence to poor local optima.

MADH thus offers a simple, scalable way to learn price-aware strategies when prices are endogenously determined by interacting agents, and it provides a practical testbed for contrasting decentralized learning with a centralized planner in applications such as prosumer energy platforms.

1.2 Application

We apply *Multi-Agent Deep Hedging* to *prosumer electricity trading platforms* (PEPs), a policy-backed market design that enables households and small businesses with distributed energy resources (photovoltaics, batteries) to trade their surplus and cover their deficits on a digital marketplace. EU legislation explicitly defines and enables peer-to-peer trading and energy sharing for active customers.[1]

From an individual prosumer's perspective, acting on a PEP is a challenging sequential decision problem. At each time step, an agent chooses (i) battery charge/discharge and (ii) photovoltaic curtailment to minimize expected daily energy cost subject to device and feasibility constraints. Platform prices are *endogenous*: they respond to the joint production, demand, and storage decisions of all agents. The battery provides an intertemporal hedge against market stochasticity—weather-driven generation and demand variability as well as price fluctuations induced by others' actions—by shifting energy from low-price to high-price periods and buffering own net-load uncertainty over time. In practice, prosumers cannot be expected to become expert traders; they will rely on algorithms and automation to steer their assets.

Our application uses MADH as a simulation laboratory to study incentives and behavior under alternative market rules and regulatory constraints. We evaluate decentralized learning—each agent optimizes its own policy while internalizing price effects—against a centralized planner that jointly optimizes all actions. This benchmark quantifies efficiency losses from decentralization and

[1] RED II defines P2P trading as "the sale of renewable energy between market participants by means of a contract with pre-determined conditions governing the automated execution and settlement of the transaction" (Directive (EU) 2018/2001, Art. 2(18)). Recent reforms further clarify that payments for energy sharing "can either be settled directly between active customers or automated through a peer-to-peer trading platform" (Directive (EU) 2024/1711, Recital (23)).

strategic interaction and highlights a key motivation for MADH in this domain: centralized control scales poorly with the number of agents, while decentralized hedging scales well.

1.3 Experimental Findings

We validate the framework on simulated communities ranging from $N = 4$ to $N = 62$ prosumers. Our experiments yield three primary insights. First, decentralized agents successfully learn profitable hedging strategies that outperform rule-based heuristics, reducing individual costs by utilizing battery arbitrage. Second, we identify a distinct trade-off between optimality and scalability. While the centralized planner (CTCE) achieves the theoretical global optimum for small clusters, it succumbs to the curse of dimensionality as the population grows. In contrast, decentralized MADH agents maintain robust performance regardless of system size.

Crucially, our analysis uncovers the emergence of strategic market behavior. As illustrated in Figure 1, agents trained with price awareness learn to exercise market power through *capacity withholding*. By discharging batteries less aggressively during peak demand, they sustain higher clearing prices to maximize export revenue. While this behavior is individually rational and shows low regret (unilateral regret $< 1.5\%$), we find that it introduces lower aggregate social welfare, disappearing with a higher number of agents.

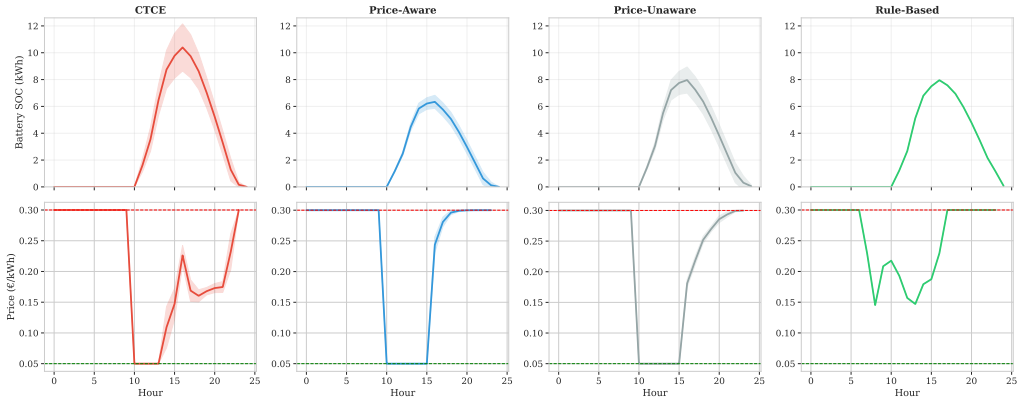


Figure 1: Average Diurnal Profiles. Top: Battery State of Charge (SoC).

1.4 Contributions.

In this work, we make the following contributions:

- **Differentiable Market Architecture:** We formulate the prosumer trading problem as a general-sum Markov game and introduce a fully differentiable market-clearing layer. Unlike standard actor-critic methods (e.g., MADDPG) that must approximate the joint Q-function, MADH allows agents to directly compute the gradient of the market price with respect to their actions.
- **Scalability Benchmark:** We provide a rigorous comparison between Centralized Training (CTCE) and Decentralized Learning (DTDE). We empirically demonstrate that while centralized planners are optimal for small N, they fail to scale ($O(e^N)$ complexity), whereas MADH agents maintain robust performance as the community grows.
- **Economic Interpretability:** We show that while strategic agents (who internalize price impact) have low regret, they achieve slightly lower total social welfare than naive price-takers.

Related Literature

This paper is situated at the intersection of multiple strands of literature. As discussed above, we build on the development of Deep Hedging for the hedging of financial portfolios in (Buehler et al., 2019) and related works. We also add to an existing literature that demonstrates the effectiveness

of MARL for managing local energy resources under uncertainty, particularly for load scheduling, storage coordination, and price-sensitive behavior in microgrids (e.g., Roesch et al., 2020; Samadi et al., 2020; Gao et al., 2021). However, this literature emphasizes system-level energy costs or improved voltage control and typically does not study strategic behavior of individual agents.[2] Examples include the use of Multi-Agent Deterministic Policy Gradient (MADDPG) to learn bidding strategies (Samende et al., 2022) and Factored Multi-Agent Centralized Policy Gradients (FACMAC), which uses a factored critic to improve scalability while managing network voltage constraints (Charbonnier et al., 2025). Feng and Liu (2025), for instance, propose a consensus-based MARL algorithm where agents share and average their critic network parameters to cooperatively align their policies and ensure network feasibility. Our work employs neither a Centralized Learning and Decentralized Execution approach nor requires direct agent communication.

Existing Multi-Agent Reinforcement Learning (MARL) approaches often rely on the centralized training and decentralized execution (CTDE) paradigm, such as MADDPG (Lowe et al., 2017) or FACMAC (**?**). These methods depend on learning a centralized critic to approximate the joint action-value function. In contrast, MADH bypasses the need for a learned critic. By modeling the market clearing as a known, differentiable operator, we propagate exact analytical gradients from the clearing price to the agent's policy. This approach aligns closer to the Deep Hedging framework (Buehler et al., 2019) than to standard model-free MARL.

In addition, there is an extensive literature on the design and taxonomy of prosumer trading markets (e.g., Khorasany et al., 2018; Sousa et al., 2019; Rodrigues et al., 2020; Zhou et al., 2020; Domènech Monfort et al., 2022; Bukar et al., 2023; Hönen et al., 2023a; Tushar et al., 2021; Tsaousoglou et al., 2022; Tushar et al., 2023). Our analytical framework adds to this literature by providing a large-scale simulation tool to investigate how decentralized policies emerge and perform under varying market designs and regulatory conditions. Our findings also contribute to the ongoing discussions on the governance of energy platforms (Weiller and Pollitt, 2013; Rosen and Madlener, 2016; Kloppenburg and Boekelo, 2019; Capper et al., 2022).

The strategic behavior of agents that we document is also connected to the literature on algorithmic collusion by sellers (e.g., Calvano et al., 2020; Eschenbaum et al., 2022), in particular on online marketplaces (e.g., Brero et al., 2022). This literature focuses on price-setting in stylized oligopoly games with standard learners (Q-learning in particular), while we study quantity-setting in a tailored application to energy platforms with a novel MADH architecture.

The paper is structured as follows. Section 3 introduces the Multi-Agent Deep Hedging (MADH) architecture. Section 2 introduces our application of prosumer trading as a POMG due to local observations and unknown states. Section 4 outlines the computational framework, including data generation and training procedures. Section 5 presents our experimental findings. Finally, Section 6 concludes.

2 Prosumer Trading Game

We apply our MADH architecture to peer-to-peer (P2P) trading on prosumer electricity trading platforms. We model the market-clearing, price formation, and policy-learning mechanisms as a partially observable Markov game (Littman, 1994; Eschenbaum et al., 2022) in which prosumer agents interact over a finite horizon. Throughout, all energy-related variables are measured in kilowatt-hours (kWh) per trading period. Because we discretize time into hourly intervals, 1 kWh is both a flow (one hour of power) and the unit of stock for battery capacity.

2.1 Partially-Observable Markov Game

Formally, the game is given by

$$\mathcal{G} = \left(\mathcal{P}, \mathcal{T}, \{\mathcal{A}^i\}, \{\mathcal{O}^i\}, \mathcal{S}, \mathcal{T}^{\text{state}}, \{r^i\}\right), \tag{1}$$

with agents $i \in \mathcal{P}$, periods $t \in \mathcal{T} = \{1, \ldots, T\}$, joint state $s_t \in \mathcal{S}$, and stochastic transition kernel $\mathcal{T}^{\text{state}}\big(s_{t+1} \mid s_t, a_t\big)$. Because each agent observes only a private signal $o_t^i \in \mathcal{O}^i$, the game features imperfect information. Policies $\pi^i : (o_{1:t}^i, a_{1:t-1}^i) \mapsto a_t^i$ must therefore cope with non-stationarity induced by the learning behavior of others. Our framework models a general-sum game or mixed

[2] Though MARL has previously been applied to automate agent bidding strategies in P2P trading environments (e.g., Zhang et al., 2018b,a; Qiu et al., 2021).

game (e.g. see (Zhang et al., 2021) without imposing restrictions on the agents' objectives (Hu and Wellman, 2003; Littman et al., 2001).

2.2 Agents

We assume that there are N prosumers, $\mathcal{P} = \{1, \dots, N\}$, each endowed with photovoltaics (PV) and a battery of capacity $\overline{B^i} \in \mathbb{R}_+$. At the beginning of period t, agent i learns her stochastic PV generation $g_t^i \in [0, \overline{G^i}]$, stochastic inelastic demand $d_t^i \in [0, \overline{D^i}]$, and current battery state $B_t^i \in [0, \overline{B^i}]$. She chooses a battery charge (positive) or discharge (negative) $b_t^i \in [-B_t^i, \overline{B^i} - B_t^i]$ and a curtailment share $\kappa_t^i \in [0, 1]$. Battery dynamics follow

$$B_{t+1}^i = B_t^i + b_t^i\big(\mathbf{1}_{\{b_t^i \geq 0\}}\eta^{\text{ch}} - \mathbf{1}_{\{b_t^i < 0\}}(\eta^{\text{dis}})^{-1}\big), \quad \eta^{\text{ch}}, \eta^{\text{dis}} \in (0, 1], \tag{2}$$

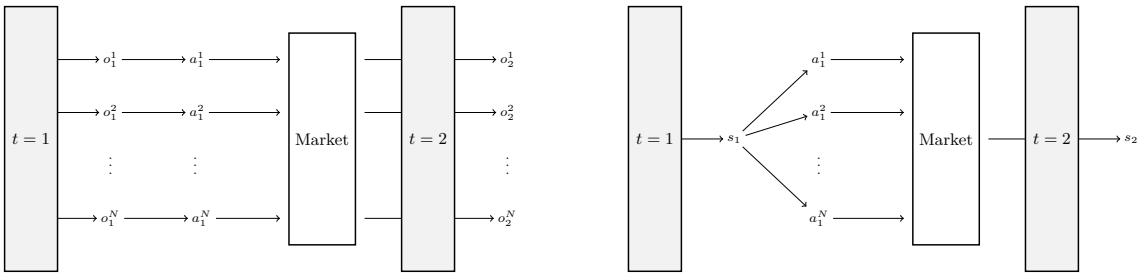
and net injection to the platform is $x_t^i = d_t^i + b_t^i - (1 - \kappa_t^i)g_t^i$, positive for imports and negative for exports[3]. Each agent observes (B_t^i, d_t^i, g_t^i, t) but neither others' battery states nor their chosen actions, making $\mathcal{G}$ a partially observable Markov game (POMG).

2.3 Environment

The platform aggregates injections, $S_t = \sum_{i=1}^N \max\{-x_t^i, 0\}$ and $D_t = \sum_{i=1}^N \max\{x_t^i, 0\}$, and posts volume-weighted sell and buy prices p_t^S and p_t^D (denoted in €/kWh) that satisfy the corridor $p^E \leq p_t^S \leq p_t^D \leq p^I$, where p^E and p^I are exogenous feed-in and retail tariffs.[4] Any admissible clearing rule is a mapping $f : (\mathbf{x}_t, p^E, p^I) \mapsto (p_t^S, p_t^D)$ with residual imbalances settled against the grid. Agent i's one-period cost c_t^i and total cost C^i are

$$c_t^i = \begin{cases} x_t^i\, p_t^D, & x_t^i > 0, \\ x_t^i\, p_t^S, & x_t^i < 0, \end{cases} \qquad C^i = \sum_{t=1}^{T} c_t^i, \tag{3}$$

respectively. Figure 2 shows the interaction of the two algorithms DTDE and CTCE with the market model over one period of the game.



(a) Decentralized Algorithm (DTDE). (b) Centralized Algorithm (CTCE).

Figure 2: Interaction of the decentralized and centralized algorithms with the market model.

2.4 Pricing Mechanism

For our experiments, we adopt the Supply-to-Demand Ratio (SDR) based pricing mechanism (Liu et al., 2017). This mechanism is simple to implement and creates dynamic prices that are inversely proportional to market supply, satisfying a basic economic principle.

The SDR at time t is defined as the ratio of the total available power supply to the total inelastic power demand within the community. Using the agent definitions from Section 2.2 (where $b_t^i > 0$ is charging and $b_t^i < 0$ is discharging), we define SDR_t as:

$$SDR_t = \frac{\sum_{i \in \mathcal{P}} \left((1 - \kappa_t^i)g_t^i - \mathbf{1}_{\{b_t^i \leq 0\}} b_t^i\right)}{\sum_{i \in \mathcal{P}} d_t^i + \mathbf{1}_{\{b_t^i \geq 0\}} b_t^i} \tag{4}$$

[3] See e.g., Lara et al. (2018); Guerrero et al. (2020); Hönen et al. (2023b)

[4] Note that participation constraints for agents imply the price corridor, as across jurisdictions prosumers are always allowed to draw from the grid at the retail price and sell to the grid at the feed-in tariff.

Here, the numerator represents the total power available (curtailed generation plus net battery discharge), and the denominator is the total inelastic demand.

The platform's buying and selling prices, p_t^D and p_t^S, are calculated as a function of SDR_t and the external grid tariffs p^I (retail/import) and p^E (feed-in/export). These prices fluctuate within the bounds of the grid tariffs, as shown by the formulas adapted from (**?**):

$$p_t^S = \begin{cases} \dfrac{p^E p^I}{(p^I - p^E)SDR_t + p^E}, & 0 \leq SDR_t \leq 1 \\ p^E, & SDR_t > 1 \end{cases} \tag{5}$$

$$p_t^D = \begin{cases} p_t^S \cdot SDR_t + p^I(1 - SDR_t), & 0 \leq SDR_t \leq 1 \\ p^E, & SDR_t > 1 \end{cases} \tag{6}$$

This mechanism ensures that the platform prices are bounded by the grid tariffs:

$$p^E \leq p_t^S \leq p_t^D \leq p^I \tag{7}$$

This dynamic pricing structure creates strong incentives for intertemporal arbitrage. When SDR_t is high (abundant supply), prices are driven down towards the feed-in tariff p^E, incentivizing agents to charge their batteries. Conversely, when SDR_t is low (scarce supply), prices rise towards p^I, incentivizing agents to discharge their batteries and sell to the platform.

3 Multi-Agent Deep Hedging Architecture

We propose Multi-Agent Deep Hedging (MADH), a method designed for finite-horizon, partially observable Markov Games (POMGs) (see Section 2.1). MADH is formulated as a multi-agent reinforcement learning (MARL) framework, and we introduce two key variants: (i) a Decentralized Training and Decentralized Execution (DTDE) setting, in which each agent learns an independent policy and acts autonomously based on their local observations, and (ii) a Centralized Training and Centralized Execution (CTCE) setting, where a single global policy is trained using full access to the global state which outputs joint actions for all agents.

Our architecture is motivated by the simplicity and proven success of MARL in industrial applications (e.g., Mnih et al., 2015; Silver et al., 2016). However, standard RL algorithms often struggle in environments characterized by high stochasticity. These challenges are amplified in MARL settings due to increased complexity and non-stationarity (Lowe et al., 2017). While our MADH architecture leverages environmental stochasticity to encourage exploration—akin to techniques used in continuous control RL methods such as DDPG (Lillicrap et al., 2015)—this passive approach to exploration is often insufficient. Therefore, explicit stochastic policies are also incorporated into MADH to enhance exploration and robustness.

3.1 Decentralized Training and Execution

In the DTDE setting, each agent $i \in \mathcal{P}$ maintains its own policy $\pi^{\theta^i} : \mathcal{O}^i \to \mathcal{A}^i$, which maps local observations to actions. These policies are parameterized as neural networks and trained independently using only local information. Agents do not observe the global state, coordinate with others, or share information—neither during training nor execution.

The policies π^{θ^i} consist of a feedforward neural network with L hidden layers and ReLU activations, followed by a tanh:[5]

$$h^0 = o^i \in \mathcal{O}^i, \quad h^\ell = \sigma\big(W_\ell h^{\ell-1} + b_\ell\big), \quad \pi^{\theta^i}(x) = \tanh\big(W_L h^{L-1} + b_L\big), \quad \ell = 1, \ldots, L-1 \tag{8}$$

where $\{W_\ell, b_\ell\}$ are agent-specific trainable parameters. Recurrent alternatives such as LSTMs may also be used in partially observable settings.

Each agent acts independently based on its private observations. These actions are submitted to a central market mechanism, which computes prices and updates the environment. The resulting

[5]The tanh function is used to bind the network's output to a specific range. In our application, this continuous output corresponds to two actions: the charging/discharging rate and the percentage of PV production to curtail.

global state induces new observations for the next individual agent decisions. Figure 3.

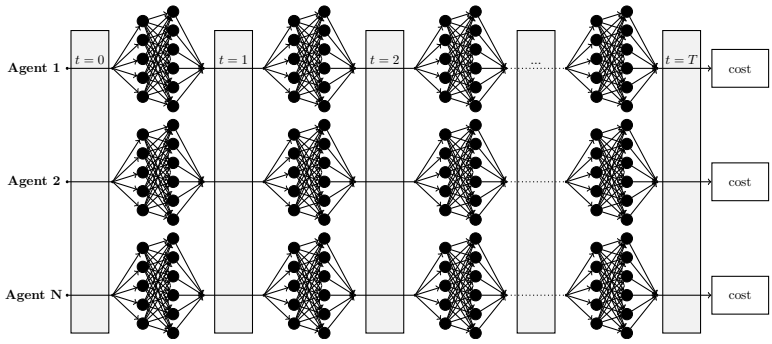


Figure 3: Each agent has an individual neural network policy that sequentially maps its private observations to actions. The transition functions (shown in grey) determine the observations available to each agent.

Policies are trained batch-wise via backpropagation through the full unrolled trajectory of the environment. We distinguish two cases in DTDE: (i) backpropagation with full computation graph, where agents engage in gradient descent through the environment updates (price-aware agents), and (ii) backpropagation without full computation graph, where agents only consider their action and not its influence on the environment (agents not being price-aware). Each trajectory records observations, actions, and incurred costs, or

$$\tau_b^i = \left\{(o_{t,b}^i, a_{t,b}^i, c_{t,b}^i)\right\}_{t=0}^{T}, \tag{9}$$

where $c_{t,b}^i$ is the cost incurred by agent i at time t in batch b. After collecting a batch, the average cost is

$$\bar{C}^i \;=\; \frac{1}{B}\sum_{b=1}^{B}\sum_{t=0}^{T} c_{t,b}^i. \tag{10}$$

To model different market behaviors, we crucially rely on controlling the flow of gradients during backpropagation. We distinguish between agents that internalize their impact on market prices versus those that treat prices as exogenous. This distinction is implemented by manipulating the automatic differentiation computation graph.

Let $\mathrm{sg}(\cdot)$ denote a stop-gradient operator (commonly implemented as a '.detach()' operation in frameworks like PyTorch). This operator evaluates its argument normally during the forward pass but blocks gradient flow during the backward pass, i.e., $\nabla_\theta\, \mathrm{sg}(z(\theta)) = \mathbf{0}$.

Price-aware (Strategic) Agents. Price-aware agents learn to internalize their market power. During training, their loss calculation uses the fully differentiable price function $p_{t,b}(\theta) = „f(\mathbf{x}_{t,b}(\theta), p^E, p^I)$. Because the market clearing function f is differentiable, gradients propagate from the agent's cost, through the pricing mechanism, back to the agent's actions. This allows the policy to learn how increasing supply might lower the market price, thereby affecting the revenue generated by that supply.

Their batch objective is:

$$\bar{C}_{\mathrm{PA}}^i(\theta) \;=\; \frac{1}{B}\sum_{b=1}^{B}\sum_{t=0}^{T} c^i\big(o_{t,b}^i, a_{t,b}^i(\theta^i), p_{t,b}(\theta)\big), \tag{11}$$

where $\theta = (\theta^1, \ldots, \theta^N)$ collects all agents' parameters. The gradient update is:

$$\theta^i \;\leftarrow\; \theta^i - \alpha\, \nabla_{\theta^i} \bar{C}_{\mathrm{PA}}^i(\theta). \tag{12}$$

Price-unaware (Naive) Agents. Price-unaware agents act as traditional price-takers, assuming their individual actions do not influence aggregate market outcomes. We enforce this behavior by detaching the calculated market prices from the computation graph before they enter the agent's

cost function. The agent sees the price value, but the connection to the actions that caused that price is severed in the backward pass.

We define the detached price as $\tilde{p}_{t,b} = \mathrm{sg}\big(p_{t,b}(\theta)\big)$. The batch objective becomes:

$$\bar{C}^i_{\mathrm{PU}}(\theta^i) \;=\; \frac{1}{B}\sum_{b=1}^{B}\sum_{t=0}^{T} c^i\big(o^i_{t,b}, a^i_{t,b}(\theta^i), \tilde{p}_{t,b}\big). \tag{13}$$

Because of the stop-gradient operator, $\nabla_{\theta^i}\tilde{p}_{t,b} = \mathbf{0}$. The agent's policy update only accounts for the direct effect of its actions on its quantity-driven costs, holding prices fixed:

$$\theta^i \;\leftarrow\; \theta^i - \alpha\,\nabla_{\theta^i}\bar{C}^i_{\mathrm{PU}}(\theta^i). \tag{14}$$

Exploration is encouraged via randomization in actions. We use a decaying ϵ-greedy scheme in which agents sample a random action uniformly from their action space with probability ϵ and otherwise follow their policy output. This ensures broad exploration early in training and shifts toward exploitation over time. Algorithm 1 summarizes training for the DTDE setting.

Algorithm 1 MADH with Decentralized Learning & Execution (DTDE)

Input: Agents $\{1, \ldots, N\}$, horizon T, batch size B, episodes K, learning rate α
for each episode $k = 1$ to K **do**
  Update probability of random action p_ε
  Sample B days in parallel:
  for training horizons $b = 1$ to B **do**
    Initialize state s_0, set cumulative costs $C^i_b \leftarrow 0$
    for $t = 0$ to T **do**
      Each agent i computes $a^i_t = \pi^{\theta^i}(o^i_t)$
      With probability ρ_k replace each component of a^i_t with random draws $\varepsilon \sim U[A]$
      Market clears: $(s_{t+1}, \{o^i_{t+1}\}) \leftarrow f(s_t, (a^1_t, \ldots, a^N_t), \xi_{t+1})$
      Market clearing f computes costs c^i_t and prices
      Accumulate cost: $C^i_b \mathrel{+}= c^i_t$
    end for
  end for
  Compute gradients and update parameters
  if price-aware *then*
    $\theta^i \leftarrow \theta^i - \alpha\,\nabla_{\theta^i}\bar{C}^i_{\mathrm{PA}}(\theta)$
  else (price-unaware)
    $\theta^i \leftarrow \theta^i - \alpha\,\nabla_{\theta^i}\bar{C}^i_{\mathrm{PU}}(\theta^i)$
end for

3.2 Centralized Learning and Execution

As an efficiency benchmark, we implement a central optimizer with full observability of all agents' states. A single policy $\pi^\theta : \mathcal{S} \to \mathcal{A}_1 \times \cdots \times \mathcal{A}_N$ is trained to minimize the total cost, $\sum_{i=1}^{N} C^i$, using batch backpropagation. The architecture of the planner mirrors that of the decentralized agents but unlike the individual agents it receives the complete state at each time period. The central optimizer then outputs actions for all agents.

Again, a decaying ϵ-greedy exploration strategy is used during training to mitigate the risk of converging to poor local minima and to promote robustness in continuous action spaces. Importantly, this central optimizer does not maximize individual agent objectives, but rather the aggregate total economic welfare of all agents.[6]

[6]Note that this central optimizer differs from the standard concept of a central planner in economic models. In typical formulations, a central planner selects the allocation of resources to all agents in order to maximize social welfare. In our case, the central optimizer instead selects the players' actions in order to maximize total welfare.

Algorithm 2 MADH with Centralized Learning & Execution

Input: Number of agents N, horizon T, batch size B, episodes K, learning rate α
for each episode $k = 1$ to K **do**
  Update probability of random action p_ε
  Sample B days in parallel:
  for training horizons $b = 1$ to B **do**
    Initialize global state s_0, set cumulative cost $C_b \leftarrow 0$
    for $t = 0$ to T **do**
      Compute joint action $a_t = \pi^\theta(s_t)$
      With probability ρ_k, replace each component of a_t with a random variable $\varepsilon \sim U[A]$
      Market clears: $s_{t+1} = f(s_t, a_t, \xi_{t+1})$
      Accumulate cost: $C_b \mathrel{+}= \sum_{i=1}^{N} c_t^i$
    end for
  end for
  Update policy: $\theta \leftarrow \theta - \alpha \nabla_\theta C_b$
end for

Figure 4 shows a centralized optimizer in which a single neural network receives the complete system state per period and outputs actions for all agents.

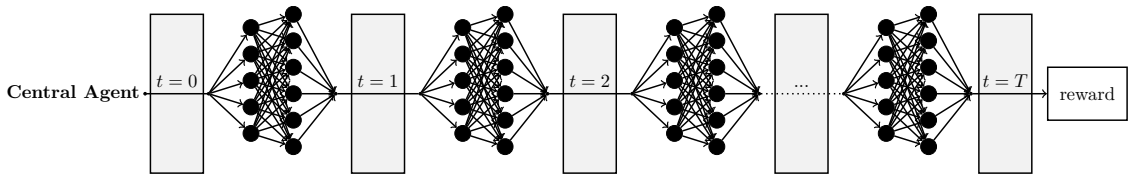


Figure 4: Centralized optimizer with joint action planning.

4 Experimental Design

We empirically validate the MADH framework within a simulated peer-to-peer (P2P) energy market populated by heterogeneous prosumers. We employ synthetic data calibrated to physical models, ensuring our results are robust to the stochastic nature of renewable generation and demand while avoiding overfitting to specific historical weather datasets.

4.1 Data Generation

We model a daily trading horizon $T = 24$ with hourly resolution. The environmental state is governed by two primary stochastic processes: photovoltaic (PV) generation and inelastic household demand.

Stochastic PV Generation. Solar generation g_t^i is modeled as a physics-based clear-sky curve scaled by a stochastic factor $\xi_t^{\text{PV}} \in [0, 1]$. To capture the temporal correlation of weather systems, the scaling factor follows a mean-reverting AR(1) process:

$$\xi_t^{\text{PV}} = \mu + \phi(\xi_{t-1}^{\text{PV}} - \mu) + \epsilon_t, \quad \epsilon_t \sim \mathcal{N}(0, \sigma^2) \tag{15}$$

where $\mu \sim U[0.6, 0.95]$ represents the daily mean irradiance. To simulate transient cloud cover, we superimpose random Gaussian pulses (negative deviations) of varying width and depth. This forces agents to learn hedging strategies against sudden intraday supply shocks.

Stochastic Demand. Inelastic demand d_t^i follows a diurnal sinusoidal baseline exhibiting characteristic morning and evening peaks. We introduce intra-day variability via a secondary AR(1) process and inject random short-duration pulses to model high-power appliance usage (e.g., EV charging). Additionally, total daily demand is log-normally distributed ($\sigma = 0.2$) across episodes to simulate seasonal occupancy changes.

4.2 Agent Heterogeneity

We investigate distributional effects by simulating a heterogeneous community. Agents are uniformly assigned one of four distinct profiles, creating a diverse ecosystem of active and passive participants:

1. *Balanced Prosumer:* Equipped with moderate generation (4 kWp) and battery storage (12 kWh), capable of active arbitrage to cover local demand.
2. *High-Generation Prosumer:* Features identical storage (12 kWh) but higher generation capacity (8 kWp), typically acting as a net seller.
3. *Solar-Only Prosumer:* Possesses PV generation (5 kWp) but lacks storage capabilities. These agents are fully exposed to spot price volatility and cannot time-shift supply.
4. *Pure Consumer:* Characterized solely by inelastic demand (20 kWh/day) with no generation or storage, acting as a passive price-taker.

4.3 Policy Architectures

We implement two distinct neural architectures to contrast decentralized and centralized learning paradigms.

Decentralized Policy Network. We parameterize the local policy π_θ for each agent i. The network input is a state vector $s_t^i \in \mathbb{R}^4$ comprising normalized features: battery state of charge (SoC), time index $t/24$, current demand d_t^i, and PV generation g_t^i. The architecture consists of two hidden layers (16 units each, ReLU activations). The output layer utilizes a tanh activation to produce continuous control signals for the battery charge/discharge rate and the curtailment ratio. These raw outputs in $[-1, 1]$ are passed through a differentiable dynamics layer to enforce physical constraints (e.g., battery capacity limits).

Centralized Policy Network. As a benchmark for the theoretical optimum, we employ a global planner that jointly determines actions for all N agents. The input is a global state vector $S_t \in \mathbb{R}^{1+3N}$, formed by concatenating the time step with the tuple (SoC, Demand, PV) for every agent. To ensure sufficient representational capacity, the network width scales with the population size ($16 \times N$ units per layer). The output layer of size $2N$ simultaneously produces specific battery and curtailment actions for the entire fleet.

Rule-Based Baseline. We compare learning-based approaches against a heuristic "self-consumption" strategy. This logic prioritizes local utilization of PV generation to minimize grid interaction, disregarding price arbitrage. During surplus ($g_t^i > d_t^i$), excess energy charges the battery to capacity before selling to the market. Conversely, during deficits, the battery discharges to cover net load, purchasing from the grid only when storage is depleted.

4.4 Training Protocol

The system is trained over $K = 50$ episodes, where each episode comprises a learning phase (50 days) followed by an evaluation phase. We optimize policies using Adam ($\alpha = 10^{-3}$, batch size $B = 32$). To facilitate exploration in the continuous action space, we employ an ϵ-greedy strategy where ϵ decays exponentially from 0.5; with probability ϵ, the agent executes a random action sampled uniformly from the valid control space. We apply gradient clipping (norm 1.0) to stabilize backpropagation.

To ensure statistical robustness, all reported results are averaged over 10 independent runs using distinct random seeds. While training data is generated stochastically to prevent overfitting, all models are evaluated on a fixed, pre-generated test set of unseen scenarios. This ensures that all algorithms are benchmarked against the exact same environmental conditions.

Baselines. We compare MADH against two boundary conditions that bracket the performance of any learning algorithm:

1. *Centralized Oracle (CTCE):* A planner with global observability. This represents the theoretical lower bound on system costs but is computationally intractable for large N.
2. *Rule-Based Heuristic:* A rigorous "maximise self-consumption" logic commonly deployed in commercial home energy management systems (HEMS). This serves as the industry-standard baseline.

We omit generic independent learners (e.g., IPPO) as our Price-Unaware (DTDE-PU) agent effectively represents this class of algorithms (treating the environment as stationary and prices as exogenous).

4.5 Evaluation Metrics

Total System Cost. We measure global market efficiency via the aggregate daily cost of all agents:

$$C_{\text{total}} = \sum_{i=1}^{N} \sum_{t=1}^{T} c_t^i \tag{16}$$

Lower total costs indicate superior resource coordination (e.g., peak-shaving via battery discharge) and reduced reliance on the external grid.

Unilateral Deviation Regret. To assess the stability of the decentralized policies, we evaluate individual regret. A set of policies constitutes an NE if no agent can significantly reduce costs by unilaterally altering its strategy. We quantify this via the unilateral deviation regret $\mathcal{R}^i$. For each agent i, we fix the trained policies of the population π^*_{-i} and retrain agent i to discover its best-response policy π^i_{BR}. The regret is the relative cost improvement achieved by this deviation:

$$\mathcal{R}^i = \frac{C^i(\pi^*) - C^i(\pi^i_{\text{BR}})}{|C^i(\pi^*)|} \tag{17}$$

where $C^i(\pi)$ denotes the cumulative cost for agent i under policy π. A regret value $\mathcal{R}^i < \varepsilon$ with ε low confirms that agents have no incentive for different policies.

5 Experimental Findings

We evaluate the Multi-Agent Deep Hedging (MADH) framework across four dimensions: (i) convergence and stability, (ii) economic efficiency and emergent behavior, (iii) scalability, and (iv) equilibrium stability. We compare three learning paradigms against a heuristic baseline:

- *CTCE (Centralized Oracle):* A global planner maximizing total social welfare (ideal benchmark).
- *DTDE–PA (Price-Aware):* The proposed method, where agents internalize their impact on the market clearing price via differentiable trading.
- *DTDE–PU (Price-Unaware):* Decentralized agents that treat prices as exogenous, ignoring their own market power.
- *Rule-Based:* A myopic controller that maximizes self-consumption (charges when $g_t > d_t$, discharges when $g_t < d_t$).

All experiments simulate heterogeneous communities of size $N \in \{4, \dots, 62\}$.

5.1 Convergence

Figure 5 illustrates the training progression of the total system cost for a community of $N = 4$ prosumers. All learning-based methods achieve stable convergence within 50 episodes, consistently outperforming the rule-based baseline.

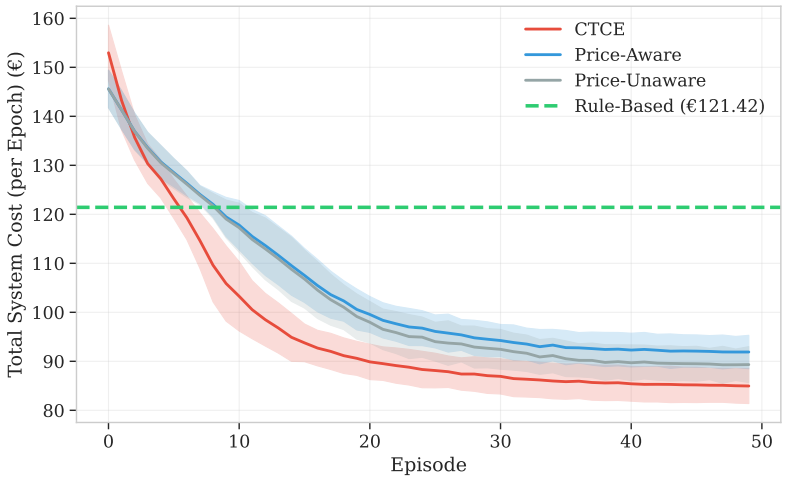


Figure 5: Convergence of Total System Cost ($N = 4$). The Centralized benchmark (Red) defines the lower bound. Notably, naive Price-Unaware agents (Grey) achieve lower total system costs than strategic Price-Aware agents (Blue).

The results highlight a trade-off between individual rationality and global efficiency. The centralized planner (*CTCE*) achieves the lowest system cost (6.1% improvement over baseline) by perfectly coordinating assets to eliminate inefficient cross-trading. However, among the decentralized methods, the naive *DTDE–PU* agents (3.5% improvement) actually outperform the strategic *DTDE–PA* agents (1.5% improvement) in terms of aggregate social welfare.

Price-Unaware agents act as competitive price-takers, clearing the market with maximum volume. In contrast, Price-Aware agents learn to exercise *market power*: they strategically withhold capacity to influence prices in their favor. While this maximizes individual utility under equilibrium constraints (see Sec. 5.4), it creates a deadweight loss that reduces the total efficiency of the system.

5.2 Strategic Battery Usage and Price Formation

To inspect the mechanism behind this efficiency gap, we analyze the diurnal battery and price profiles in Figure 6.

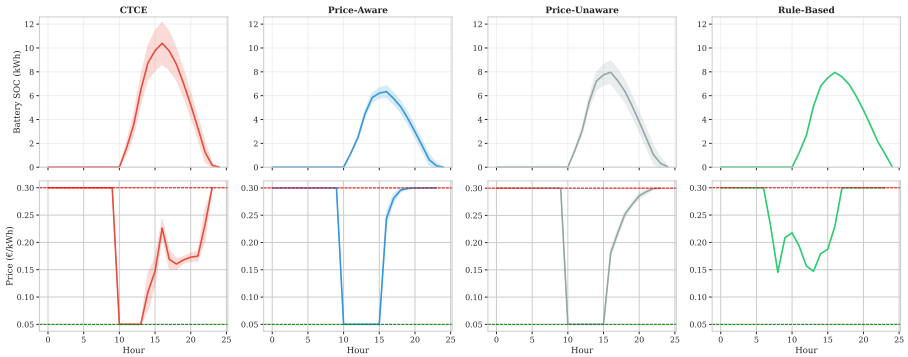


Figure 6: Average Diurnal Profiles ($N = 4$). Top: Battery SoC. Bottom: Market Prices. Price-Aware agents (Blue) exhibit strategic behavior—discharging less aggressively during evening peaks than Price-Unaware agents (Grey)—to sustain favorable selling prices.

The *DTDE–PU* agents (Grey) behave aggressively: they charge during solar peaks and discharge rapidly during evening demand (18:00–20:00), causing sharp price jumps. In contrast, *DTDE–PA* agents (Blue) learn a *capacity withholding* strategy. They utilize less battery capacity and discharge gradually over a longer horizon. By rationing their supply, they prevent the clearing price from crashing, effectively manipulating the market to maximize the value of their exports

rather than the volume. *CECT* uses more capacity than the other regimes and has generally lower prices in the evening period.

This state-dependent strategy is visualized in Figure 7. The policy heatmap confirms that Price-Aware agents learn to charge the battery when PV generation is abundant, with the discharging region improving with a higher SOC.

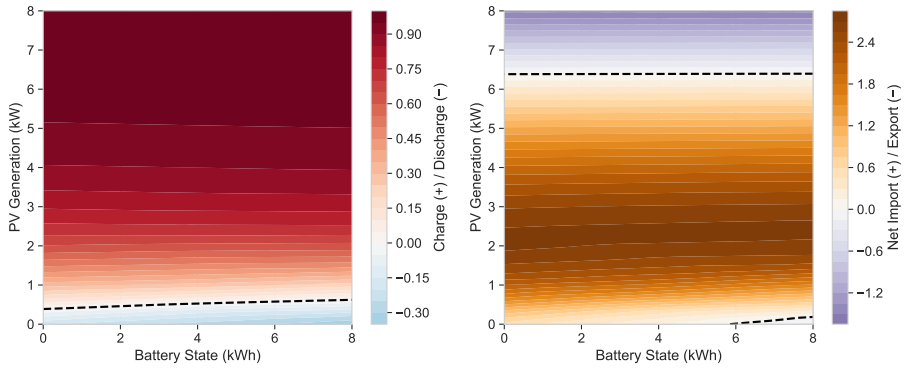


Figure 7: Policy Heatmap (DTDE–PA). The agent learns a gradient strategy: charging (light) when PV is high/SoC is low, and discharging (dark) as SoC increases.

5.3 Scalability and Computational Trade-offs

We benchmark scalability by increasing the community size N. Figure 8 (Left) reveals the computational cost of strategic learning. *DTDE–PA* training time grows quadratically, as it requires backpropagating gradients through the market-clearing mechanism for all agents ($O(N^2)$ interactions). Conversely, *DTDE–PU* and *CTCE* remain computationally efficient per episode.

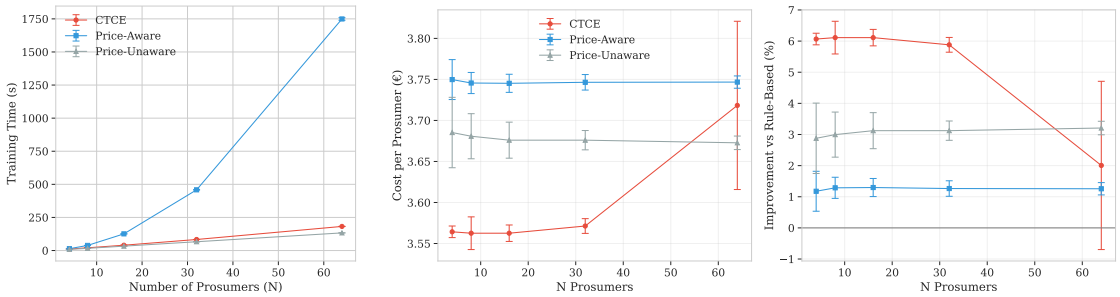


Figure 8: Scalability ($N \in \{4, \ldots, 60\}$). Left: Training time per episode. Price-Aware learning is computationally intensive. Middle and right: Cost optimality. The centralized planner (Red) fails to scale, degrading in performance as N grows, while decentralized agents maintain stable performance.

Figure 8 (Left) reveals the computational cost of strategic learning. DTDE–PA training time grows quadratically ($O(N^2)$), as the backward pass requires propagating gradients through the clearing mechanism where every agent's action affects the price faced by every other agent. Conversely, DTDE–PU scales linearly ($O(N)$) as price gradients are detached. However, the Centralized Planner (CTCE) exhibits a different failure mode. While its training time per step is acceptable, the *sample complexity* required to explore the joint action space $\mathcal{A}^1 \times \cdots \times \mathcal{A}^N$ grows exponentially. This results in the performance degradation seen in Figure 8 (Right), where the centralized policy fails to converge to an optimal solution within the fixed computational budget for $N = 60$.

5.4 Regret Analysis

Finally, we verify the stability of the learned strategies by measuring *ex-post regret*—the cost reduction an agent could achieve by unilaterally deviating to a best-response policy. For the *DTDE–*

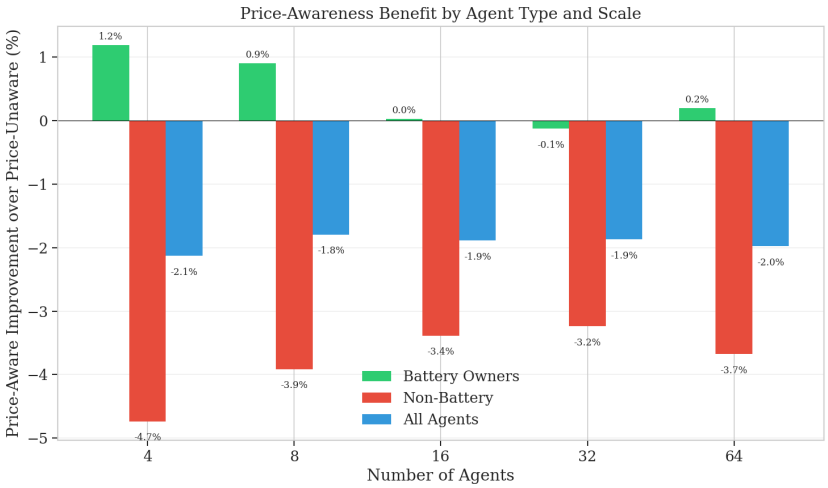


Figure 9: Relative Improvement of Price-Aware vs. Price-Unaware. The benefit of strategic awareness is highest in small markets (oligopoly) and decays as N increases (competitive limit).

PA population ($N = 8$), average regret is negligible ($< 1.5\%$). Figure 10 confirms this stability: the best-response trajectory (dashed) deviates only marginally from the learned equilibrium policy (solid), yielding minimal gain.

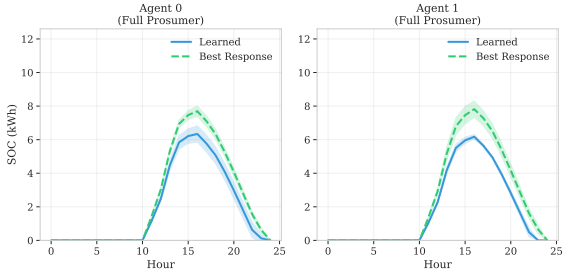


Figure 10: Equilibrium Stability. The learned DTDE–PA policy (Solid) is nearly identical to the theoretical Best Response (Dashed).

6 Conclusion

This work introduces Multi-Agent Deep Hedging (MADH), a scalable framework for learning decentralized trading policies in markets with endogenous price formation. By embedding a differentiable market-clearing mechanism directly into the computation graph, we enable agents to learn complex, non-linear strategies that internalize their impact on market prices without requiring explicit communication or central coordination.

Our empirical application to peer-to-peer energy trading reveals critical insights into the deployment of autonomous agents in economic systems. First, we demonstrate that scalability is a decisive advantage of the decentralized approach. While the centralized welfare-maximizing planner (CTCE) provides a theoretical lower bound on costs for small communities, it succumbs to the curse of dimensionality as the population grows. In contrast, decentralized MADH agents maintain robust performance regardless of system size.

Second, our results quantify the Price of Anarchy in algorithmic markets. We find that strategic (Price-Aware) agents show low ex-post regret. However, this individual rationality comes at a cost to social welfare: strategic agents learn to withhold capacity to support prices, resulting in lower aggregate efficiency compared to naive (Price-Unaware) agents. This highlights a fundamental tension: while DTDE-UA agents yield higher total welfare, their policies are exploitable; conversely, strategic agents form stable markets but introduce deadweight loss through oligopolistic behavior.

Ultimately, MADH serves as a powerful simulation testbed for market design. It allows regulators and platform operators to anticipate the emergent behaviors of market agents—such as capacity withholding or peak shifting—and iteratively refine market rules to align individual incentives with system-level efficiency.

Limitations and Theoretical Considerations. A central assumption of MADH is the differentiability of the market clearing function $f(\cdot)$. While this holds for Supply-to-Demand Ratio (SDR) and smoothed approximations of order books, it may not directly apply to discrete matching mechanisms (e.g., strict merit-order stacks) without relaxation techniques. Furthermore, while our regret analysis confirms convergence with low regret, theoretical convergence guarantees for policy gradient methods in general-sum non-convex games remain an open research challenge. Our work relies on empirical validation of stability rather than analytical proofs.

Future Work. While this study focused on economic trade-offs, real-world deployment requires physical feasibility. Future work will extend MADH to incorporate grid constraints (e.g., voltage and line limits) directly into the differentiable clearing mechanism. Additionally, investigating the interplay between MADH agents and legacy grid controllers offers a promising avenue for designing hybrid regulatory frameworks that balance market freedom with grid stability.

References

Brero, G., Mibuari, E., Lepore, N., and Parkes, D. C. (2022). Learning to mitigate ai collusion on economic platforms. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 37892–37904. Curran Associates, Inc.

Buehler, H., Gonon, L., Teichmann, J., and Wood, B. (2019). Deep hedging. *Quantitative Finance*, 19(8):1271–1291.

Bukar, A. L., Hamza, M. F., Ayup, S., Abobaker, A. K., Modu, B., Mohseni, S., Brent, A. C., Ogbonnaya, C., Mustapha, K., and Idakwo, H. O. (2023). Peer-to-peer electricity trading: A systematic review on currents development and perspectives. *Renewable Energy Focus*.

Calvano, E., Calzolari, G., Denicolò, V., and Pastorello, S. (2020). Artificial intelligence, algorithmic pricing, and collusion. *American Economic Review*, 110(10):3267–97.

Capper, T., Gorbatcheva, A., Mustafa, M. A., Bahloul, M., Schwidtal, J. M., Chitchyan, R., Andoni, M., Robu, V., Montakhabi, M., Scott, I. J., et al. (2022). Peer-to-peer, community self-consumption, and transactive energy: A systematic literature review of local energy market models. *Renewable and Sustainable Energy Reviews*, 162:112403.

Charbonnier, F., Peng, B., Vienne, J., Stai, E., Morstyn, T., and McCulloch, M. (2025). Centralised rehearsal of decentralised cooperation: Multi-agent reinforcement learning for the scalable coordination of residential energy flexibility. *Applied Energy*, 377:124406.

Curin, N., Kettler, M., Kleisinger-Yu, X., Komaric, V., Krabichler, T., Teichmann, J., and Wutte, H. (2021). A deep learning model for gas storage optimization. *Decisions in Economics and Finance*, 44(2):1021–1037.

Domènech Monfort, M., De Jesús, C., Wanapinit, N., and Hartmann, N. (2022). A review of peer-to-peer energy trading with standard terminology proposal and a techno-economic characterisation matrix. *Energies*, 15(23):9070.

Eschenbaum, N., Mellgren, F., and Zahn, P. (2022). Robust algorithmic collusion. *arXiv preprint arXiv:2201.00345*.

Feng, C. and Liu, A. L. (2025). Peer-to-peer energy trading of solar and energy storage: A networked multiagent reinforcement learning approach. *Applied Energy*, 383:125283.

Gao, Y., Wang, W., and Yu, N. (2021). Consensus multi-agent reinforcement learning for volt-var control in power distribution networks. *IEEE Transactions on Smart Grid*, 12(4):3594–3604.

Guerrero, J., Gebbran, D., Mhanna, S., Chapman, A. C., and Verbič, G. (2020). Towards a transactive energy system for integration of distributed energy resources: Home energy management, distributed optimal power flow, and peer-to-peer energy trading. *Renewable and Sustainable Energy Reviews*, 132:110000.

Hönen, J., Hurink, J. L., and Zwart, B. (2023a). A classification scheme for local energy trading. *OR Spectrum*, 45(1):85–118.

Hönen, J., Hurink, J. L., and Zwart, B. (2023b). Dynamic rolling horizon-based robust energy management for microgrids under uncertainty. *arXiv preprint arXiv:2307.05154*.

Hu, J. and Wellman, M. P. (2003). Nash q-learning for general-sum stochastic games. *Journal of machine learning research*, 4(Nov):1039–1069.

Khorasany, M., Mishra, Y., and Ledwich, G. (2018). Market framework for local energy trading: A review of potential designs and market clearing approaches. *IET Generation, Transmission & Distribution*, 12(22):5899–5908.

Kloppenburg, S. and Boekelo, M. (2019). Digital platforms and the future of energy provisioning: Promises and perils for the next phase of the energy transition. *Energy Research & Social Science*, 49:68–73.

Krabichler, T. and Teichmann, J. (2023). A case study for unlocking the potential of deep learning in asset-liability-management. *Frontiers in Artificial Intelligence*, 6:1177702.

Lara, J. D., Olivares, D. E., and Canizares, C. A. (2018). Robust energy management of isolated microgrids. *IEEE Systems Journal*, 13(1):680–691.

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

Littman, M. L. (1994). Markov games as a framework for multi-agent reinforcement learning. In *Machine learning proceedings 1994*, pages 157–163. Elsevier.

Littman, M. L. et al. (2001). Friend-or-foe q-learning in general-sum games. In *ICML*, volume 1, pages 322–328.

Liu, N., Yu, X., Wang, C., Li, C., Ma, L., and Lei, J. (2017). Energy-sharing model with price-based demand response for microgrids of peer-to-peer prosumers. *IEEE Transactions on Power Systems*, 32(5):3569–3583.

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

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. (2015). Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

Qiu, D., Wang, J., Wang, J., and Strbac, G. (2021). Multi-agent reinforcement learning for automated peer-to-peer energy trading in double-side auction market. In *IJCAI*, pages 2913–2920.

Rodrigues, D. L., Ye, X., Xia, X., and Zhu, B. (2020). Battery energy storage sizing optimisation for different ownership structures in a peer-to-peer energy sharing community. *Applied Energy*, 262:114498.

Roesch, M., Linder, C., Zimmermann, R., Rudolf, A., Hohmann, A., and Reinhart, G. (2020). Smart grid for industry using multi-agent reinforcement learning. *Applied Sciences*, 10(19):6900.

Rosen, C. and Madlener, R. (2016). Regulatory options for local reserve energy markets: Implications for prosumers, utilities, and other stakeholders. *The Energy Journal*, 37(2_suppl):39–50.

Samadi, E., Badri, A., and Ebrahimpour, R. (2020). Decentralized multi-agent based energy management of microgrid using reinforcement learning. *International Journal of Electrical Power & Energy Systems*, 122:106211.

Samende, C., Cao, J., and Fan, Z. (2022). Multi-agent deep deterministic policy gradient algorithm for peer-to-peer energy trading considering distribution network constraints. *Applied Energy*, 317:119123.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., et al. (2016). Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489.

Sousa, T., Soares, T., Pinson, P., Moret, F., Baroche, T., and Sorin, E. (2019). Peer-to-peer and community-based markets: A comprehensive review. *Renewable and Sustainable Energy Reviews*, 104:367–378.

Tsaousoglou, G., Giraldo, J. S., and Paterakis, N. G. (2022). Market mechanisms for local electricity markets: A review of models, solution concepts and algorithmic techniques. *Renewable and Sustainable Energy Reviews*, 156:111890.

Tushar, W., Nizami, S., Azim, M. I., Yuen, C., Smith, D. B., Saha, T., Poor, H. V., et al. (2023). Peer-to-peer energy sharing: A comprehensive review. *Foundations and Trends® in Electric Energy Systems*, 6(1):1–82.

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

Weiller, C. M. and Pollitt, M. G. (2013). *Platform markets and energy services*. JSTOR.

Zhang, K., Yang, Z., and Basar, T. (2018a). Networked multi-agent reinforcement learning in continuous spaces. In *2018 IEEE conference on decision and control (CDC)*, pages 2771–2776. IEEE.

Zhang, K., Yang, Z., and Başar, T. (2021). Multi-agent reinforcement learning: A selective overview of theories and algorithms. *Handbook of reinforcement learning and control*, pages 321–384.

Zhang, K., Yang, Z., Liu, H., Zhang, T., and Basar, T. (2018b). Fully decentralized multi-agent reinforcement learning with networked agents. In *International conference on machine learning*, pages 5872–5881. PMLR.

Zhou, Y., Wu, J., Long, C., and Ming, W. (2020). State-of-the-art analysis and perspectives for peer-to-peer energy trading. engineering 6 (7): 739–753.