

Exploring Market Designs for Enhanced Flexibility Procurement with Deep Reinforcement Learning

Viktor Zobernig^{1,2}, Sarah Fanta¹, Stefan Strömer^{1,2}, Regina Hemm¹,
Jochen Stiasny², Jochen L. Cremer^{1,2}, Laurens J. de Vries²

¹AIT Austrian Institute of Technology, Vienna, ²Delft University of Technology, The Netherlands
email: viktor.zobernig@ait.ac.at

Abstract—The growing share of renewable energy in short-term European electricity markets has significantly increased congestion management costs and demands. Therefore, current market design is not optional to keep congestion costs low. A proper market would incentivize the integration of flexibilities to boost competition and lower costs, while mitigating risks of manipulation. However, assessing behavioral impacts is challenging due to increasingly interconnected market structures. Studies modeling more than two markets often overlook the strategic opportunities that emerge from these interactions, focusing instead on large-scale dynamics. To capture the detailed impact of bidding strategies, we use reinforcement learning to explore multi-market strategies. By progressively training a Deep Reinforcement Learning (DRL) agent as a market participant—from replicating established behaviors to mastering intricate multi-market interactions—we employ Domain-Informed Curriculum Learning (DomCL), a structured approach that systematically guides learning through staged complexity. We validate our approach against established two-market studies, then evaluate it in two progressively complex four-market case studies spanning a 6-bus network, including historical data. Results show that our DRL-based method improves performance while uncovering challenges that arise as strategic opportunities expand, offering a structured approach for multi-market design analysis.

Index Terms—Agent-Based Model, Deep Reinforcement Learning, Electricity Markets, Market Power, Strategic Bidding

I. INTRODUCTION

The increasing share of renewable energy sources, combined with ongoing market integration and the slow pace of grid expansion, has introduced significant challenges to the existing European electricity market design. During periods of high renewable generation, electricity production often exceeds the transmission network’s capacity. The prevailing zonal market-clearing mechanism is ill-equipped to manage these bottlenecks, resulting in a growing reliance on redispatch measures. This reliance drives up operational costs and hinders potential reductions in CO₂ emissions, adding further complexities to market operations. Furthermore, this situation creates opportunities for market manipulation, such as the so-called “inc-dec gaming” — a strategy that exploits congestion management mechanisms by overstating offered volumes or underbidding marginal costs, knowing that these bids are unlikely to be activated. The potential introduction of market-based redispatch remuneration mechanisms raises concerns about amplifying such behavior, as participants could leverage higher price bids to account for increased opportunity costs associated with upward regulation.

Challenges to Capturing Multi-Market Behavior: Addressing the challenges of modern electricity markets requires market design improvements that incentivize flexibility integration while mitigating market manipulation. Simulating strategic bidding behavior is essential for evaluating market design assessing price formation, identifying flaws, and ensuring effectiveness. However, conventional analytical methods face inherent limitations with scalability, perfect foresight assumptions, reliance on predefined probability distributions, and handling non-convexities—all essential for computational feasibility. These constraints restrict the integration of multiple markets, agents, or time steps, significantly affecting the representation of strategic opportunities and risks.

To model strategic behavior, conventional approaches determine Nash equilibria by formulating payoff functions, as in game theory, or by modeling each market participant as an optimization problem constrained by others’ equilibrium conditions—formalized as EPEC (Equilibrium Problem with Equilibrium Constraints) and MPEC (Mathematical Program with Equilibrium Constraints). Notable examples include [1], which demonstrates systemic risks and dynamics of inc-dec gaming, offering insights in the dominant strategy even under perfect competition in market-based redispatch. [2], [3] apply EPEC models to study local market power in inc-dec gaming. Collectively, these studies focus on single time steps within two-market setups. While effective in identifying dominant strategies in equilibrium scenarios, their reliance on nested optimization problems leads to exponential growth in computational intensity with increasing decision points, making them impractical for larger-scale or more dynamic settings. These limitations often introduce biases in modeling bidding behavior, which weakens the interpretation of market outcomes. Similarly, the assumption of perfect foresight limits these models’ to capture the uncertainty and complexity of real-world interactions.

In contrast, stochastic optimization (SO) addresses hidden information by accounting for risks and uncertainties, avoiding the assumption of perfect foresight and being less constrained by scalability. For instance, [4]–[6] use stochastic optimization to assess the profitability of market participants across multiple markets, highlighting both the method’s potential and its limitations. However, in doing so, SO relies on predefined probability distributions, which limits its ability to capture evolving interactions across multiple markets or competitors.

Beyond analyzing strategic decisions at the micro level,

large-scale models often focus on macro-level dynamics, such as sector coupling or system-wide interactions [7], [8]. These models simplify strategic bidding into rule-based decisions to manage computational complexity. More advanced approaches, like the MASCEM model, consider (Deep) reinforcement learning (DRL) to select the most effective predefined strategy, combining adaptability with heuristic rules [9], [10]. However, these works do not model strategies behavior at finer resolutions which is important for capturing realistic market outcomes.

DRL has been successfully applied to various challenges in strategic interaction [11], [12]. [13] investigated DRL for finding Nash equilibria, [14], [15] applied RL for joint bidding and pricing, and [16], [17] used multi-agent RL to study day-ahead market dynamics. Additionally, [18] incorporated ramp-up and ramp-down costs into DRL models to handle non-convexities. Despite these advancements, research on multi-market decision-making remains limited and often focuses only on interactions between two markets [19]–[21], as increasing the number of markets introduces computational intractability. A key research gap lies in modeling strategic bidding behavior across multiple markets, where growing complexity leads to additional constraints and trade-offs over various time horizons—challenges that standard Markov Decision Process (MDP) formulations in DRL applications may fail to capture [22].

Proposed Contribution: Motivated by this gap, we examine why standard MDP formulations in DRL fail to capture cross-market dependencies and dynamic constraints in multi-market scenarios. We then propose domain-informed curriculum learning (DomCL) to systematically introduce sub-scenarios reflecting real-world complexities. By leveraging domain knowledge, DomCL enables agents to establish foundational behaviors before addressing the full complexity of the model, providing a more scalable solution for multi-market integrations. We demonstrate the scalability of this approach by comparing it to standard implementations of a DRL algorithm and validating it against a well-established example from the literature. Furthermore, through two case studies spanning four markets (Balancing Capacity and Energy, Day-ahead, and Redispatch), we evaluate the use of DomCL to handle systematically increasing environmental complexity by analyzing how the DRL agent identifies and exploits strategies such as inc-dec gaming. Finally, as numeric methods remain challenged in producing feasible solutions in complex market scenarios, we demonstrate that DRL, guided by DomCL, effectively uncovers optimal strategies. This underscores its potential for modeling strategic decision-making in multi-market settings and supports a structured approach to analyzing market design and behavior across interconnected markets.

We outline the challenges of standard MDP formulation in Sec. II-A and introduce DomCL as a solution in Sec. II-B. Section II-C describes its integration with our DRL-based Flexibility Service Provider (FSP) agent. In Sec. III-A, we describe the market environment before presenting our findings in Sec. III-B. We conclude in Sec. IV. Appendices A to D provide additional details on the algorithm and data.

II. ADDRESSING EFFECTIVE DRL IMPLEMENTATION IN MULTI-MARKET SYSTEMS

In this section we first examine DRL’s challenges in multi-market settings, then present DomCL as a solution and outline the DRL-based FSP model. For details on reinforcement learning, see [23].

A. Challenges in Multi-Market Modeling with DRL

In DRL, the optimal policy π^* is obtained through iterative updates using dynamic programming, which models the environment as a MDP. An MDP defines transitions from a state s_t to the next state s_{t+1} with the agent taking an action a_t , and receiving a reward r_t , determined by the reward function $R(s_t, a_t)$ [24]. The main objective is to maximize the expected return G_t , defined as the discounted cumulative reward. This is achieved using the Bellman equation, which relates the action-value function $Q_\pi(s_t, a_t)$ to the expected return under policy π :

$$Q_\pi(s_t, a_t) = \mathbb{E}_\pi[R(s_t, a_t) + \gamma Q_\pi(s_{t+1}, a_{t+1}) | s_t, a_t] \quad (1)$$

When considering M markets, the agent sequentially places bids $a_{t,m}$ in each market state $s_{t,m}$, where future market states depend on past actions $s_{t,M}(s_{t,0:M-1}, a_{t,0:M-1})$. Here, t indexes episodes (each representing a full day of market interactions), while m indexes sequential market interactions within an episode. However, policy updates in standard MDPs only consider immediate rewards, ignoring dependencies beyond adjacent steps and leading to myopic decision-making. For example, if the state at market $s_{t,m+3}$ also depends on $s_{t,m}$ and $s_{t,m+1}$, the MDP fails to capture this dependency. Instead, the agent optimizes $s_{t,m}$ based solely on $s_{t,m+1}$ disregarding the long-term effect on $s_{t,m+3}$. Figure 1 visually illustrates this limitation. If an action yields short-term benefits in $s_{t,m}$ but negatively impacts $s_{t,m+3}$, the agent is not penalized unless explicitly modeled. As a result, the policy update only accounts for immediate effects.

To address this constraint, different strategies have been proposed. For instance, [25], [26] reformulated an autoregressive Q-function to model dependencies across multiple steps. While these approaches implicitly encode information from past observations, they have only shown success when state dynamics between interrelated actions are negligible or when trained offline on large datasets.

A more formal alternative to handle partial observability of past states is by extending the problem to a Partially Observable Markov Decision Process (POMDP) [22]. A POMDP generalizes the MDP by maintaining a belief state—a probability distribution over possible underlying states, conditioned on

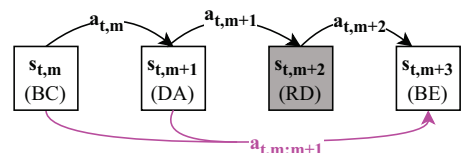


Fig. 1. State dependencies in an MDP. The final state is influenced only by the preceding action from the previous state (gray), while decisions from earlier stages (pink) are ignored.

observations and the MDP dynamics. Unlike an MDP policy, which directly maps states to actions, a POMDP’s policy maps a history of observations (or belief states) to actions. This allows dependencies across non-adjacent markets to be captured, ensuring that $s_{t,M}(s_{t,0:M-1}, a_{t,0:M-1})$ holds.

To address policy training with long-term state dependencies, curriculum learning [27] provides a structured framework by incrementally training sub-policies. This constrains the solution space, systematically capturing how actions influence long-term rewards and improving the agent’s ability to model complex dependencies across sequential decisions.

B. Domain-Informed Curriculum Learning

To capture long-term dependencies of bidding decisions across sequential markets, we adopt a curriculum learning strategy and represent the market process as a POMDP. The learning process for the optimal policy π^* is partitioned into sequential sub-tasks, where each sub-policy π_k is trained on a subset of markets, leveraging knowledge from previous stages. This structured approach enables a systematic mapping of states (or belief states) to actions, ensuring the agent incrementally learns inter-market dependencies rather than attempting to model all potential market interactions at once. Instead of randomly learning from an extensive number of possible market-state relationships, we enhance training efficiency by systematically incorporating domain knowledge. The agent first learns strategies for the balancing markets (BC and BE), followed by redispatch (RD), and finally optimizes bidding across all four markets. Each stage includes the day-ahead market (DA) and is illustrated in Fig. 2 within the “Market Environment” diagram.

To define which markets are active at each stage, we introduce a binary proxy vector $\mathbf{p} \in \{0, 1\}^M$, where M is the total number of markets. At each market m , $\mathbf{p}_{t,m}$ contains entries of 1 for active markets and 0 for inactive ones.

During stage k , the agent’s policy $\pi_k(a_{t,m} | b_{t,m}, \mathbf{p}_{t,m})$ is optimized to maximize the cumulative reward,

$$\pi_k^* \approx \mathbb{E} \left[\sum_{t=0}^T \gamma^t R(s_{t,m}, a_{t,m}) \mid b_{t,m}, \mathbf{p}_{t,m} \right], \quad (2)$$

where $b_{t,m}$ is the belief state, capturing structured dependencies across markets.

When transitioning from stage k to stage $k + 1$, the policy is updated to incorporate additional market dependencies:

$$\pi_{k+1}(a_{t,m} | b_{t,m}, \mathbf{p}_{t,m}) = \pi_k(a_{t,m} | b_{t,m}, \mathbf{p}_{t,m}) + \Delta\pi_k(a_{t,m}), \quad (3)$$

where $\Delta\pi_k(a_{t,m})$ modifies the policy as an update step, progressively integrating new dependencies while preserving previously learned strategies $k + 1$. By leveraging sub-policies from earlier stages, the agent retains learned market dependencies, incrementally incorporating new interactions without forgetting prior knowledge. The binary proxy vector $\mathbf{p}_{t,m}$ implicitly encodes the belief state for the policy during training, effectively guiding the agent’s focus towards relevant market interactions at each stage. This structured encoding further facilitates reward penalization by allowing for delayed feedback, where the overall achieved profits reflect the

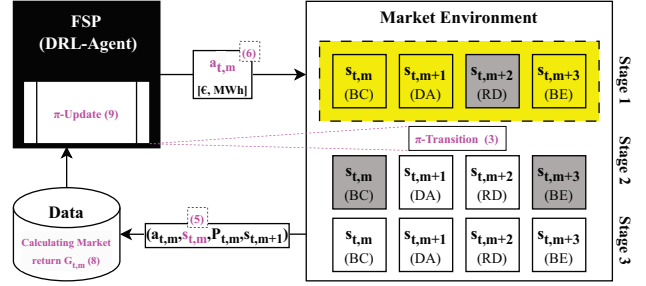


Fig. 2. Illustration of the FSP agent’s training process using DomCL in a four-market environment. Grayed-out market states indicate omitted markets at different stages, while the yellow box highlights the active market interaction. Pink elements highlight the policy update process, reward post-processing, and FSP interaction with the model environment, with references to key equations from the text

long-term consequences of bidding decisions, as detailed in Sec. II-C.

C. Flexibility Service Provider - DRL Agent Setup

We introduce a formal model for the FSP, which exploits market opportunities by bidding strategically to maximize profits based on available market information. The implemented DRL logic follows the Twin-Delayed Deep Deterministic Policy Gradient (TD3) algorithm [28], a *model-free* approach that eliminates the need to assume prior knowledge of the environment’s dynamics. Furthermore, it adapts its policy through *online* learning, directly interacting with the environment in real-time. TD3 employs an actor-critic architecture, where the actor (policy network) outputs deterministic actions. In this setup, the FSP, modeled using TD3, observes a market state $s_{t,m}$ and selects an action $a_{t,m}$, representing a bidding decision (including price and volume), using the policy function $\pi_\theta(s_{t,m})$, parameterized by θ :

$$\pi_\theta(s_{t,m}) = a_{t,m} \quad (4)$$

Below, we define the state information provided to the agent, the bidding actions it can execute, and the profit calculation mechanism. Details of the algorithm’s update process, including the hyperparameters used during training, the historical data utilized, and a summary of the complete FSP agent, are provided in Appendix A.

States: The FSP’s decision at each market m for each day t depends on the following state variables:

$$s_{t,m} = [C_{t,m}^{avail}, C_{t,m}^{run}, CP_{t,m}^{hist}, W_{t,m}, \mathbf{p}_{t,m}] \quad (5)$$

- $C_{t,m}^{avail}$: Available capacity.
- $C_{t,m}^{run}$: Running electricity (used capacity).
- $CP_{t,m}^{hist}$: Past clearing prices (last 3 days).
- $W_{t,m}$: Weather forecasts (of the current day).
- $\mathbf{p}_{t,m}$: Market proxy vector, indicating active markets. (see Sec. II-B for further details).

Each state component influences specific aspects of the decision-making process: $C_{t,m}^{avail}$ and $C_{t,m}^{run}$ inform volume bids, ensuring they remain within feasible operational limits, while $CP_{t,m}^{hist}$ and $W_{t,m}$ guide the pricing strategy, allowing

the agent to adapt to expected price fluctuations. All data, except for the *Market Proxy*, are in hourly resolution.

Actions: We assume a single divisible bid per hour (similar to the Colombian bidding format), requiring the agent to decide both volume and price. Thus, actions are represented as a $1 \times (2H)$ vector, comprising H price and H volume bids per market.

$$a_{t,m} = [VB_{t,m}^{1:H}, PB_{t,m}^{1:H}] \quad (6)$$

where $VB_{t,m}^h$ and $PB_{t,m}^h$ represent volume and price bids for each hour in each market. A hyperbolic tangent activation function maps all actions to the range $[-1, 1]$, where:

- $VB_{t,m}^h$: $1 = \max(C_{t,m}^{avail})$, $-1 = -\max(C_{t,m}^{run})$.
- $PB_{t,m}^h$: $1 = \text{Price cap}$, $-1 = \text{Price floor}$.

Rewards: The FSP optimizes its bidding policy based on (2), where the reward corresponds to the profits $P_{t,m}$ calculated as the total profit from each hour h at market m of the current day t , excluding fixed costs:

$$R_{t,m} = P_{t,m} = \sum_h^H SC_{t,m}^h * (CP_{t,m}^h - MC_{t,m}^h) \quad (7)$$

where $SC_{t,m}^h$ is the sold capacity, $CP_{t,m}^h$ the clearing price and $MC_{t,m}^h$ is the marginal cost. To capture long-term dependencies across markets, the reward function is defined using the return $G_{t,m}$ over markets, which accumulates discounted future rewards:

$$G_{t,m} = \sum_{w=0}^{M-m} \gamma^w P_{t,m+w} \quad (8)$$

where $\gamma \in [0,1)$ is the discount factor. Since TD3 is an *off-policy* algorithm, data is stored in a replay buffer before updates, allowing reward modifications without violating the assumption that the agent cannot access future information during training. This effectively aligns reward penalization with $\mathbf{p}_{t,m}$, which implicitly encodes the belief state.

Policy Update: The policy is updated iteratively based on the expected return:

$$\pi_{\theta}^* \approx \mathbb{E} \left[\sum_{t=0}^T \gamma^t G_{t,m} \right] \quad (9)$$

An overview of the FSP's interaction with the market environment is provided in Fig. 2, highlighting the exchange of state information, bidding actions, return calculation, and policy updates specific to this study.

III. RESULTS

To assess the performance of the implemented agent setup, we investigate an EPEC model from the literature [2], applied to a two-market setup testing the impact of the research gap. We then evaluate our DRL agent to bid strategically in an extended four-market environment through two progressively complex case studies: (I) the *Four-Markets Baseline* case and (II) the *Four-Markets Flex+* case, designed to assess its strengths and limitations.

A. Market Environment

We adopt the environment from [2]: based on a six-bus (from [29]) network including loads, transmission lines, and generators across DA and RD markets we test our approach. To enhance the market setting, we extend this environment to include BC and BE markets, enabling four-market experiments. Congestion is managed through a two-step linear power flow (LOPF) process: first, the DA market clears without line capacity constraints to identify overloaded lines. Then, a second LOPF run incorporates line limits and RD bids while fixing the DA dispatch.. To better capture real-world variability, we further extended the static two-market model by integrating historical weather and load data, introducing non-Gaussian stochasticity. Historical data simulates grid imbalances in balancing markets. Accepted positive capacity bids are reserved, while negative bids are mandatorily placed at the minimum price in the DA market and restricted for RD in the *Four-Markets Baseline*. BE bids are auto-generated from reserved capacity, leaving only price decisions. As stated in Sec. II-C, we assume a single, divisible bid is submitted per hour. To reduce computational complexity, data is divided into four six-hour blocks, with data points sampled accordingly. Additionally, the data is scaled to match the magnitude of the six-bus model. For a detailed description of the data and preprocessing methods, refer to Appendices C and D.

B. DRL Performance in Multi-Market Environment

The main objective of this study is to apply DRL to develop a scalable approach for assessing bidding strategies in sequential electricity markets. Before extending to the four-market case, we first benchmark our method against the EPEC model in a two-market system from [2]. Unlike the EPEC model, which fixes decisions on volume bids, our approach operates in a continuous action space, allowing dynamic bidding decisions, setting volume as well as price choices. For a fair comparison, we constrained the agent's actions to match the EPEC model. Under these conditions, the agent converged to the same equilibrium bidding strategy, achieving 99.8% of the profits obtained in the EPEC model. The majority of these profits stem from the agent's ability to identify and exploit in-dec gaming strategies, replicating the exact behavior observed in the EPEC model.

I. Four-Markets Baseline: Expanding to four markets allows for a more rigorous evaluation of DRL's scalability in capturing strategic FSP behavior under increasing market complexity. However, training without DomCL leads to a significant decline in performance, as evidenced by lower total profits (see Fig. 3). To ensure a fair comparison between training with and without DomCL, we match the total number of training iterations for both approaches. The results show that DomCL significantly improves enhances the agent's performance, with the final training *Stage 3* (blue boxplot) yielding considerably higher rewards than the baseline training without DomCL. The black dashed line represents the mean profit. Additionally, the staged progression of DomCL reveals distinct differences in reward distributions: balancing markets (*Stage 1*, light grey box) exhibit lower profit potential than RD markets

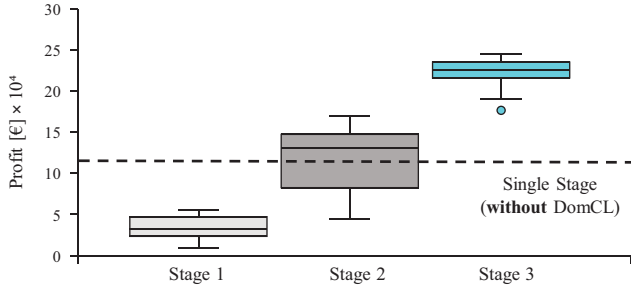


Fig. 3. *Four-Markets Baseline*: Comparison of achieved profits in the last 50 episodes of each training stage using DomCL (boxplots) against the mean profit from training without DomCL (black dashed line).

(*Stage 2*, dark grey box), where the agent retains local market power and can exploit inc-dec gaming more effectively.

The observed disparity stems from how the agent formulates its decision-making strategy under different training setups. Without DomCL, the agent treats the problem as an MDP, adopting a weighted strategy that averages optimal decisions across markets based on their frequency. As a result of MDP-based decision-making, the agent follows an intermediate approach that cannot fully exploit interdependencies between markets, limiting its strategic flexibility. In contrast, DomCL enables the agent to model the problem as a POMDP, allowing it to dynamically adapt and align strategic opportunities across markets. By leveraging these observed market interdependencies, the agent formulates a more cohesive and substantially more profitable bidding strategy.

Ultimately, with DomCL, the agent adapts its strategy to broader market conditions during the final third stage. In specific situations, it mitigates the potential for reduced profits from inc-dec gaming caused by market fluctuations driven by renewable generation and DA demand, by engaging in balancing. These results demonstrate how DRL can allow the integration of multiple markets to facilitate long-term bidding decision-making under imperfect information when treating the problem as POMDP acquired by the proposed DomCL. These results show how DRL enables multi-market integration and supports long-term bidding decisions under imperfect information by treating the problem as a POMDP through the proposed DomCL.

II. Four-Markets Flex+: This case study evaluates the robustness of DomCL in training multi-market strategies for FSPs by reinforcing the interdependence between policies learned in Stages 1 and 2, which form the foundation for the final strategy in Stage 3. To test this, we introduce a novel market design where the TSO can deploy capacity procured from the BC market for both balancing and RD. This increased flexibility may incentivize market participants to offer additional capacity due to higher activation probabilities while fostering greater competition and potentially lowering overall costs. However, as the DRL agent retains local market power, its strategic behavior remains unchanged, resulting in no significant impact on market prices. While one might expect the agent to hedge its inc-dec gaming gains more easily by participating in both flexibility markets, it instead achieves slightly lower profits than in the *Four-*

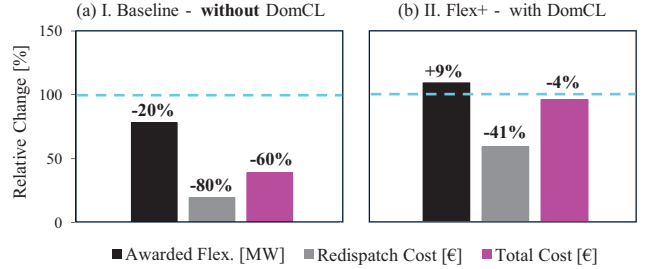


Fig. 4. Percentage ratios of awarded flexibilities (including BE and RD volumes), RD costs, and total system costs relative to the *Four-Markets Baseline* with DomCL, for (a) *Four-Markets Baseline* without DomCL and (b) *Four-Markets Flex+* with DomCL.

Markets Baseline (see Fig. 4b). Detailed results indicate that increased system flexibility, combined with weaker traceable action–reaction patterns, makes inefficiencies harder to identify. Moreover, any remaining opportunities—despite the agent’s initial market power—are more volatile, increasing risk and encouraging more conservative strategies. This suggests that participants in a regulator-controlled system (e.g., without access to hidden information) may struggle to detect or exploit residual inefficiencies.

Given this complexity, validating the advanced market designs becomes inherently challenging, especially when no clear optimum exists or can be formally defined. This underscores the rationale for using DRL, as conventional numerical optimization methods may fail to produce feasible solutions. Consequently, the best practice involves demonstrating measurable improvements over established benchmarks. Thus, Fig. 4 compares model outcomes for the previous case study without DomCL (a) and the second case study using DomCL (b), both relative to the *Four-Market Baseline*. The results highlight the significant performance gains achieved with DRL and DomCL but also reveal challenges in the more advanced *Four-Market Flex+* setup. Despite theoretically higher achievable margins, overall performance drops slightly, suggesting unresolved complexities.

IV. CONCLUSION

This study applies DRL to develop a scalable approach for assessing bidding strategies in sequential electricity markets. We introduce DomCL to enable DRL within a POMDP framework and show that the FSP agent converges to the same equilibrium bidding strategy as an established EPEC model in a two-market setup. In a four-market setting with historical data, DomCL yields notable performance gains. However, increased market flexibility dilutes market feedback, weakening the agent’s ability to detect inefficiencies and leading to more conservative bidding behavior. These findings highlight both the potential and the remaining challenges of applying DRL in multi-market environments. The proposed framework offers a structured approach for evaluating market design, and examining strategic behavior across interconnected market settings, thereby paving the way for further research on strategic decision-making.

ACKNOWLEDGMENTS

Parts of the results were obtained during the project DigI-Plat, that has received funding in the framework of the joint programming initiative ERA-Net Smart Energy Systems' focus initiative Digital Transformation for the Energy Transition, with support from the European Union's Horizon 2020 research and innovation program under grant agreement No 883973. Additionally, the position of Prof. De Vries is sponsored by the Dutch government agency EBN (www.ebn.nl).

REFERENCES

- [1] L. Hirth and I. Schlecht, "Market-Based Redispatch in Zonal Electricity Markets," Nov. 2018, USAEE Working Paper No. 18-369. [Online]. Available: <https://ssrn.com/abstract=3286798>
- [2] M. Sarfati and P. Holmberg, "Simulation and Evaluation of Zonal Electricity Market Designs," *Electric Power Systems Research*, vol. 185, p. 106372, Aug. 2020.
- [3] A. M. Systad and J. L. Eilertsen, "An analysis of mitigating measures for inc-dec gaming in market-based redispatch," Master's thesis, NTNU, 2022. [Online]. Available: <https://ntnuopen.ntnu.no/ntnu-xmlui/handle/11250/3027173>
- [4] E. Kraft, M. Russo, D. Keles, and V. Bertsch, "Stochastic optimization of trading strategies in sequential electricity markets," *European Journal of Operational Research*, vol. 308, no. 1, pp. 400–421, Jul. 2023.
- [5] K. Van Der Linden, N. Romero, and M. M. De Weerd, "Benchmarking Flexible Electric Loads Scheduling Algorithms," *Energies*, vol. 14, no. 5, p. 1269, Feb. 2021.
- [6] S. Ø. Ottesen, A. Tomasgard, and S.-E. Fleten, "Multi market bidding strategies for demand side flexibility aggregators in electricity markets," *Energy*, vol. 149, pp. 120–134, Apr. 2018.
- [7] M. Reeg *et al.*, "AMIRIS: An Agent-Based Simulation Model for the Analysis of Different Support Schemes and Their Effects on Actors Involved in the Integration of Renewable Energies into Energy Markets," in *2012 23rd International Workshop on Database and Expert Systems Applications*. Vienna, Austria: IEEE, Sep. 2012, pp. 339–344.
- [8] S. Glismann, "Ancillary Services Acquisition Model: Considering market interactions in policy design," *Applied Energy*, vol. 304, p. 117697, Dec. 2021.
- [9] Z. Vale, T. Pinto, I. Praca, and H. Morais, "MASCEM: Electricity Markets Simulation with Strategic Agents," *IEEE Intelligent Systems*, vol. 26, no. 2, pp. 9–17, Mar. 2011.
- [10] European Commission: Directorate-General for Energy, Moser, A., Bracht, N. and Maaz, A., *Simulating electricity market bidding and price caps in the European power markets – S18 report*. Publications Office, 2019. [Online]. Available: <https://data.europa.eu/doi/10.2833/252345>
- [11] O. Vinyals *et al.*, "Grandmaster level in StarCraft II using multi-agent reinforcement learning," *Nature*, vol. 575, no. 7782, pp. 350–354, Nov. 2019.
- [12] J. Perolat *et al.*, "Mastering the game of Stratego with model-free multiagent reinforcement learning," *Science*, vol. 378, no. 6623, pp. 990–996, Dec. 2022.
- [13] C. Graf, V. Zobernig, J. Schmidt, and C. Klöckl, "Computational Performance of Deep Reinforcement Learning to Find Nash Equilibria," *Computational Economics*, vol. 63, no. 2, pp. 529–576, Feb. 2024.
- [14] H. Xu *et al.*, "Deep Reinforcement Learning for Joint Bidding and Pricing of Load Serving Entity," *IEEE Transactions on Smart Grid*, vol. 10, no. 6, pp. 6366–6375, Nov. 2019.
- [15] H. Xu, Q. Wu, J. Wen, and Z. Yang, "Joint bidding and pricing for electricity retailers based on multi-task deep reinforcement learning," *International Journal of Electrical Power & Energy Systems*, vol. 138, p. 107897, Jun. 2022.
- [16] N. Harder, R. Qussous, and A. Weidlich, "Fit for purpose: Modeling wholesale electricity markets realistically with multi-agent deep reinforcement learning," *Energy and AI*, vol. 14, p. 100295, Oct. 2023.
- [17] Y. Du, F. Li, H. Zandi, and Y. Xue, "Approximating Nash Equilibrium in Day-ahead Electricity Market Bidding with Multi-agent Deep Reinforcement Learning," *Journal of Modern Power Systems and Clean Energy*, vol. 9, no. 3, pp. 534–544, 2021.
- [18] Y. Ye, D. Qiu, M. Sun, D. Papadaskalopoulos, and G. Strbac, "Deep Reinforcement Learning for Strategic Bidding in Electricity Markets," *IEEE Transactions on Smart Grid*, vol. 11, no. 2, pp. 1343–1355, Mar. 2020.
- [19] K. Poplavskaya, J. Lago, and L. De Vries, "Effect of market design on strategic bidding behavior: Model-based analysis of European electricity balancing markets," *Applied Energy*, vol. 270, p. 115130, Jul. 2020.
- [20] T. Wolgast, E. M. Veith, and A. Nieße, "Towards reinforcement learning for vulnerability analysis in power-economic systems," *Energy Informatics*, vol. 4, no. S3, p. 21, Sep. 2021.
- [21] J. Tran *et al.*, "Deep Reinforcement Learning for Modeling Market-Oriented Grid User Behavior in Active Distribution Grids," in *2021 IEEE PES Innovative Smart Grid Technologies Europe (ISGT Europe)*. Espoo, Finland: IEEE, Oct. 2021, pp. 01–06.
- [22] K. Åström, "Optimal control of Markov processes with incomplete state information," *Journal of Mathematical Analysis and Applications*, vol. 10, no. 1, pp. 174–205, Feb. 1965.
- [23] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed., ser. Adaptive Computation and Machine Learning Series. Cambridge, Massachusetts: The MIT Press, 2018.
- [24] R. Bellman, "A Markovian Decision Process," *Journal of Mathematics and Mechanics*, vol. 6, no. 5, pp. 679–684, 1957.
- [25] L. Metz, J. Ibarz, N. Jaitly, and J. Davidson, "Discrete Sequential Prediction of Continuous Actions for Deep RL," Jun. 2019. [Online]. Available: <http://arxiv.org/abs/1705.05035>
- [26] Y. Chebotar *et al.*, "Q-Transformer: Scalable Offline Reinforcement Learning via Autoregressive Q-Functions," in *7th Annual Conference on Robot Learning*, Aug. 2023. [Online]. Available: <https://openreview.net/forum?id=0I3su3mkuL>
- [27] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th Annual International Conference on Machine Learning*. Montreal Quebec Canada: ACM, Jun. 2009, pp. 41–48.
- [28] S. Fujimoto, H. Hoof, and D. Meger, "Addressing Function Approximation Error in Actor-Critic Methods," in *Proceedings of the 35th International Conference on Machine Learning*. PMLR, Jul. 2018, pp. 1587–1596.
- [29] H.-P. Chao and S. Peck, "A Market Mechanism for Electric Power Transmission," *Journal of Regulatory Economics*, vol. 10, no. 1, pp. 25–59, 1996.
- [30] D. Silver *et al.*, "Deterministic Policy Gradient Algorithms," in *Proceedings of the 31st International Conference on Machine Learning*. PMLR, Jan. 2014, pp. 387–395.
- [31] T. Akiba, S. Sano, T. Yanase, T. Ohta, and M. Koyama, "Optuna: A Next-generation Hyperparameter Optimization Framework," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. Anchorage AK USA: ACM, Jul. 2019, pp. 2623–2631.
- [32] Austrian Power Grid (APG). APG Market Data and Grid Information. Accessed: 2024-08-12. [Online]. Available: <https://www.apg.at/en>
- [33] ENTSO-E Transparency Platform. Accessed: 2024-08-12. [Online]. Available: <https://transparency.entsoe.eu/>

APPENDIX A

TD3 ALGORITHM AND HYPER PARAMETERS

A. Updating Process

The critic, or Q -network, approximates the Q -values using weights and biases θ^Q , and its loss function is defined as:

$$L = \frac{1}{N} \sum_j (y_j, Q(s_j, a_j | \theta^Q))^2 \quad (10)$$

where N is the size of the minibatch of sampled transition (s_j, a_j, r_j, s_{j+1}) from the memory buffer. The target value, y_j , is computed as:

$$y_j = r_j + \gamma Q'(s_{j+1}, \pi'(s_{j+1} | \theta^{Q'})). \quad (11)$$

Here, Q' and π' are target networks - copies of the actor and critic networks used to stabilize training by preventing harmful interdependencies during updates. These target networks are updated by using soft updates:

$$\theta^{Q'} \leftarrow \tau \theta^Q + (1 - \tau) \theta^{Q'} \quad (12)$$

$$\theta^{\pi'} \leftarrow \tau\theta^\pi + (1 - \tau)\theta^{\pi'} \quad (13)$$

The actor network, denoted as the π -network with weights θ^π , outputs deterministic actions $\pi^\theta(s_t)$, as described in [30]. Since the policy is deterministic, Gaussian noise $\mathcal{N}(0, \sigma)$ is added to the executed actions during training to ensure sufficient exploration. The actor is updated by maximizing the expected Q-value of the selected action:

$$\nabla_{\theta^\pi} J \approx \frac{1}{N} \sum_j \nabla_a Q(s, a | \theta^\pi) |_{s=s_j, a=\pi(s_j)} \nabla_{\theta^\pi} \pi(s | \theta^\pi) |_{s_j}. \quad (14)$$

To improve stability and performance, TD3 employs twin critics and delayed updates. Twin critics mitigate overestimation bias by maintaining two Q-value estimators and using their minimum for target computation. Delayed updates ensure that the actor and target networks are updated less frequently (every two critic updates), allowing the critics to stabilize before influencing the policy. The complete FSP agent, based on TD3, is outlined in Algorithm 1.

B. Applied Hyper Parameters

Hyper-parameters are initially selected through empirical testing and later fine-tuned using "Optuna" [31] for automated optimization. The final hyper-parameters applied are: $\gamma = 0.99$, $\tau = 0.995$, and a replay buffer size of $|\mathcal{R}| = 10^6$. Gaussian exploration noise is drawn from $\mathcal{N}(0, \sigma)$, initially scaled by 0.1 and linearly reduced to 0.01 throughout training. Additionally, Gaussian noise, scaled by 0.2, is also to the target y_j in (11)

The neural network architecture employs an actor learning rate of $\lambda_\pi = 1e-4$ and a critic learning rate of $\lambda_Q = 1e-3$. Both networks comprise two hidden layers with 256 neurons each, utilizing a batch size of 64. Batch normalization is applied to enhance stability. The following number of training iterations are implemented for the two-market and four-market setups:

- **Four-Market Setup:** Each training epoch, corresponding to one sampled year of data (see Appendix D), consists of 28 days, with four sequential market sessions per day. The total training duration is:
 - *With DomCL:* $N = 80$ epochs per stage.
 - *Without DomCL:* $N = 240$ epochs in total.

This results in a total of $4 \times 28 \times 80 \times 3$ training steps. During the exploration phase, the first 1000 steps are executed with random actions, and policy updates commence after 500 steps.

- **Two-Market Setup:** Here, training is performed in a single stage over one day, comprising two consecutive market sessions. The training process requires 2×1000 steps to converge. During the exploration phase, the first 300 steps are taken with random actions, and policy updates begin after 150 steps.

In both setups, policy updates are applied every four time steps, with four update steps executed per cycle.

Algorithm 1 Flexibility Service Provider Agent using DomCL

- 1: **Define:** $K = \{\mathcal{M}_1, \dots, \mathcal{M}_K\}$ as markets per stage and \mathcal{V} as the total executed market steps in stage k
- 2: Initialize O market states using random actions, and V^{start} market states before starting updates
- 3: **for** each stage k in K **do** \triangleright Each consisting of N epochs
- 4: Reset environment and obtain initial state:

$$s_{t,0} = [C_{t,0}^{avail}, C_{t,0}^{run}, CP_{t,0}^{hist}, W_{t,0}, \mathbf{P}_{t,0}]$$

- 5: Sample dataset S for training (see Appendix D)
- 6: Set counter for visited market states: $\mathcal{V} \leftarrow 0$
- 7: **for** each episode t **do**
- 8: **for** each market $m \in \mathcal{M}_k$ **do**
- 9: Increment visited market states: $\mathcal{V} \leftarrow \mathcal{V} + 1$
- 10: Select action:
$$a_{t,m} = \begin{cases} \mathcal{U}(A), & \text{if } \mathcal{V} \leq O \\ \pi_k(s_{t,m}) + \mathcal{N}_t(0, \sigma), & \text{else} \end{cases}$$
- 11: Execute action: $a_{t,m} = [VB_{t,m}^{1:H}, PB_{t,m}^{1:H}]$
- 12: Observe next state $s_{t,m+1}$ and reward $r_{t,m}$
- 13: Store transition $(s_{t,m}, a_{t,m}, r_{t,m}, s_{t,m+1})$ in \mathcal{R}
- 14: **end for**
- 15: Compute and store return $G_{t,m}$ in \mathcal{R} using:

$$G_{t,m} = \sum_{w=0}^{M-m} \gamma^w P_{t,m+w}$$

- 16: **if** $\mathcal{V} \geq V^{start}$ **then**
- 17: Sample minibatch (s_j, a_j, r_j, s_{j+1}) from \mathcal{R}
- 18: Compute critic gradient by (10), with:
$$y_j = G_j + \gamma Q'(s_{j+1}, \pi'_k(s_{j+1} | \theta^{Q'}))$$
- 19: Update critic: $\theta^Q \leftarrow \theta^Q - \lambda_Q \nabla_{\theta^Q} L(\theta^Q)$
- 20: Compute policy gradient by (14)
- 21: Update policy: $\theta^\pi \leftarrow \theta^\pi - \lambda_\pi \nabla_{\theta^\pi} J$
- 22: Update target networks by (12) and (13)
- 23: **end if**
- 24: **end for**
- 25: Policy Result from Stage k (Training over \mathcal{M}_k):

$$\pi_k^* \approx \mathbb{E} \left[\sum_{t=0}^T \gamma^t G_{t,m} \right]$$

- 26: Policy Update for Next Stage: $\pi_{k+1} \leftarrow \pi_k^* + \Delta \pi_k$
 - 27: **end for**
-

APPENDIX B

TECHNICAL DETAILS MARKET ENVIRONMENT

The market environment is based on the Chao and Peck 6-bus network [29], adapted from [2]. While the original setup included only the day-ahead (DA) and redispatch (RD) markets, we extend it by incorporating balancing capacity (BC) and balancing energy (BE) markets. An overview of the 6-bus network, including generator capacities and marginal costs, is provided in Fig. 5.

Market clearing procedures, including congestion emergence and resolution, are structured as follows

TABLE I
MARKET TIMING, PRICING RULES, AND BID FORWARDING MECHANISMS

Market	GCT	Pricing Rule	Forwarding Mechanism
Balancing Capacity	D-1, 10:00	Pay-as-Bid	Accepted bids are reserved (positive and negative).
Day-Ahead	D-1, 12:00	Pay-as-Cleared	Reserved capacities must be offered at the price floor.
Redispatch	D-1, 18:00	Pay-as-Bid	Only in Four-Markets Flex+: Reserved capacities (positive and negative) are converted into energy bids. Free bidding is allowed; single-price bids are added.
Balancing Energy	T-25 min	Pay-as-Cleared	Reserved capacities (positive and negative) are converted into energy bids. Free bidding is not allowed; only price bids are added.

- **BC Market:** Procures secondary reserves (aFRR) via LOPF at two nodes—one for upward, one for downward capacity.
- **DA Market:** Solved via LOPF without line constraints to determine initial dispatch.
- **Congestion Check:** A secondary LOPF run with fixed DA dispatch detects congestion by enforcing line limits.
- **RD Market:** If congestion occurs, nodal pricing redispatches while maintaining DA feasibility.
- **BE Market:** Clears imbalances using accepted BC reserves via LOPF without line constraints.

We use single divisible bids per hour and agent. Additional constraints for BC and BE bids are applied:

- **Capacity Bids:** Negative bids must be placed in the DA market at the price floor; positive bids are reserved for the BC.
- **Balancing Energy Bids:** Auto-generated from accepted BC bids, restricting free bidding.

The market clearing timing and forwarding mechanisms are summarized in Table I.

APPENDIX C LOAD AND WEATHER DATA

We integrate load and weather data for our four-market use cases and model redispatch needs based on resulting weather patterns and DA load fluctuations. This highlights a key advantage of DRL training—its ability to learn without requiring large amounts of historical data upfront.

A. Data Scaling

For all markets except RD, we incorporate Austrian load and price data (November 2022–October 2023) from Austrian Power Grid (APG) [32]. Weather forecasts and real-time renewable availability stem from ENTSO-E’s Transparency Platform [33].

To align historical data with the literature model, we scale the day-ahead demand using a factor α , where:

$$\mathbf{Y} = \alpha \mathbf{X}, \quad \alpha = \frac{\mu_{Model}}{\mu_{DA_Data}} \quad (15)$$

where μ_{Model} is the fixed demand from the model, and μ_{DA_Data} is the mean of the historical dataset. Price data remains unscaled.

APPENDIX D DATA SAMPLING

A uniform hourly resolution is applied across all markets. The BC market, operating in six 4-hour blocks, is interpolated to hourly values, while BE market data, originally recorded at 15-minute intervals, is aggregated to hourly samples.

To capture seasonal and weekly variations, a structured sampling approach is applied:

- *Seasonal Division:* The year is split into four seasons of 13 weeks each.
- *Weekly Selection:* One representative week is sampled per season.
- *Daily and Hourly Sampling:* Each day is divided into four 6-hour intervals, with one data point sampled per interval:
 - Interval 1: 00:00–05:59
 - Interval 2: 06:00–11:59
 - Interval 3: 12:00–17:59
 - Interval 4: 18:00–23:59

The final dataset S consists of four sampled weeks, one from each season, with seven days per week and four data points per day, ensuring representation of daily, weekly, and seasonal patterns.

