MTHE Group 13 Thesis Report: Optimizing Power Generator Bids using Q-Learning

Thomas Boyle, Natalie Chow, Sam Joffe, Rohan Kumar

20290727, 20348865, 20352098, 20280792

April 12th, 2026

Abstract

This thesis studies the problem of how a single, a few, or many energy generators should bid into a day-ahead electricity market with the goal of maximizing individual profit. Agents must develop a strategy that factors in demand variance, fuel costs, and competitive behavior. The thesis formulates this game as a finite Markov Decision Process, and uses tabular Q-learning to learn how much to bid and at what price, using historical market data. The implemented framework incorporates discretization of the market's states and agent actions, a profit based reward function that penalizes unfeasible decisions, and a data-proxied market simulator that generates heterogeneous opponent bids. The Markov Decision Process was modeled in such a way to balance the tradeoff of keeping key market features while preserving computational tractability.

Simulation results show that the Q-learning policies outperform a rule-based cost-plus markup and quantile policy, both in the single-agent and multi-agent case. In the single agent case, simulated with data from the ERCOT energy market, the Q-learning policy's mean profit per episode performed 29.2% better than the best cost-plus markup policy. In the 5 agent case, this gap widened to 79.17%. Moreover, various indicators point towards Q-value stabilization, implying the convergence of the RL agents to a learned policy, within the context of the simulator.

This thesis demonstrates that tabular Q-learning can produce competitive bidding strategies for energy generators. The simulation results suggest that as the number of competitors in a given market increase, so does the need for a sophisticated strategy, for which Q-learning is a propitious framework. This thesis also tackles the engineering tradeoff within this problem, of needing to balance realism and tractability, and consequently the need for proper validation of tractability assumptions made.

Acknowledgements

We would like to express our gratitude to our supervisor, Dr. Serdar Yüksel. Without his guidance, this project would have gone down a much less interesting path. His insights were invaluable.

We would also like to thank the team at Enverus, namely, Thomas Mulvihill, Juan Arteaga, and Adam Robinson, for our discussions, and their support and resources. Their ISO insights heavily shaped our modeling of the game, and their resources are the heart of our implementation.

Lastly, we would like to thank our friends and colleagues within the Applied Mathematics and Engineering program for their continued support and encouragement.

Contents

List of Tables

List of Figures

1 Introduction

This section gives background information on electricity markets in North America, gives a description of possible solutions to this problem, and formulates an exact statement of the bidding optimization problem discussed in this thesis.

1.1 Background / Motivation

In the United States and Canada, energy markets are managed by states and provinces rather than at a federal level. Despite there not being a unified market, there are three types of markets commonly observed: day-ahead markets, real-time markets, and capacity markets. Sometimes, a geographic location will have more than one type of market. These markets are operated by Independent System Operators (ISOs) and Regional Transmission Organizations (RTOs), which manage transmission grids and facilitate wholesale electricity markets. The remaining demand is serviced by vertically integrated organizations taking on operational, financial, and distribution responsibilities within their geographic area, which are geographically grouped as bilateral organizations[10].

Regardless of market type, most organized power markets follow a uniform clearing price auction model. Bids from generators are sorted from least to most expensive in what is called the stack. The cumulation of the cheapest bids satisfying the projected demand are dispatched, in what is referred to as clearing the market. The bid price of the next MWh increment above the projected demand is used to determine the price paid to all market participants who cleared the market[14]. Day-ahead markets allow suppliers to put in financially binding bids for each hour of the following day; they are then paid or charged based on the prices observed in the real-time market. Over time, the increasing number of market participants and growing penetration of renewable energy sources have intensified competition within these markets[32].

1.2 Problem Statement

At a high level, the challenge addressed in this thesis is the following: given that electricity markets are competitive and subject to uncertain demand, renewable generation, and competitor behaviour, how should a power generator bid in order to maximize its long-run profit? This is a problem that affects any asset owner participating in a wholesale electricity market, and it is non-trivial even for experienced market participants.

Despite the fact that North American market structure is quite advanced and uses standardized clearing price auctions [14], duality pricing methods for spot prices [20], and more recently CAPM models [33], finding an optimal bid still proves to be a difficult issue. Participants never have complete information. They must bid without knowing how demand will change, how generation from renewables will vary, or how competitors will act. A successful bidding strategy must achieve marginal cost recovery, account for competitor behaviour, and balance maximizing expected returns against managing risk. This strategy must also account for artificial competition being added to the system through additional agents in a

multi-agent setting.

There are multiple ways to solve this problem, with several methodologies already existing. Early work used to rely on rule-based and cost-plus strategies, in which generators bid at a fixed markup above their marginal cost of production, this is a fairly straight forward strategy that does not require much explanation [9]. Recently, more sophisticated approaches have used mathematical programming and game-theoretic models to account for strategic interactions among market participants [19]. Reinforcement learning methods have also been applied to electricity market bidding, with Q-learning in particular having been shown to be effective in multi-agent market settings [25]. Despite this body of work, a unified framework that combines data-driven market simulation, realistic opponent modelling, and multi-agent Q-learning in a tractable and interpretable way remains an open problem.

Formally, the bidding optimization problem can be stated as follows. Let $\mathcal{S}$ denote the set of observable market states and $\mathcal{A}$ denote the set of admissible bids. The goal is to find a policy $\pi^* : \mathcal{S} \to \mathcal{A}$ that maximizes the long-run expected discounted profit

$$\pi^* \in \arg\max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} \right],$$

subject to the following constraints and considerations:

- **Capacity constraint:** the quantity bid by any agent must not exceed its physical generation capacity, $q_t \leq q_{\max}$.
- **Risk constraint:** the notional exposure of any bid, defined as price times quantity, must not exceed a maximum allowable threshold, $p_t \times q_t \leq \text{Notional}_{\max}$.
- **Marginal cost recovery:** bids should be designed to ensure that the expected clearing price covers the agent's marginal cost of production c_i.
- **Computational feasibility:** the strategy must be computable efficiently on large historical market datasets.

In addition to the mathematical formulation, higher-level factors influence bidding strategy design. Economic constraints require strategies to not only compute maximum profit but also be computationally efficient on large market datasets such as hourly historic prices, ISO bidding history, ancillary service prices, and long-term capacity estimates. Environmental factors highlight the importance of storage in enabling the integration of renewables and the reduction of fossil generation. Social considerations concern fairness and transparency in access to the market, ensuring that strategies do not undermine trust in the system. Health and safety issues of batteries, from thermal stability to degradation, must also be noted when setting out strategies that may affect real operations.

Overall, the primary challenge is to build a framework that integrates reinforcement learning and game theory so that bidding strategies for power assets in North American electricity markets are optimized. The framework must be mathematically sound and feasible, and

adaptable to both operational and broader market situations. The MDP formulation, developed in detail in Section 2, serves as the core design solution to this problem. By resolving this problem, the project has the dual mission of enhancing profitability for asset owners while contributing to the efficiency, reliability, and sustainability of the electricity grid.

1.3 Potential Solutions

There were three main classes of approaches considered for designing bidding strategies in this project: CAPM-based optimization, simulation-based optimization, and Q-learning-based reinforcement learning.

CAPM-based optimization. In more recent years, mathematicians have been looking at how to price energy assets from a financial engineering perspective, as if they were a portfolio of stock market assets. The key link between these models is that they build on the Capital Asset Pricing Model (CAPM), a model developed for asset managers to determine if the expected return of a derivative is worth the risk involved with holding or selling it[7, 33]. For electricity assets, this idea can be extended by asking whether supplying energy at bid price B, when the production cost is C, yields a sufficiently high rate of return

$$r = \frac{B - C}{C}$$

to justify the risk σ associated with bidding at that price. When we frame the discussion this way, the bidding problem becomes one of maximizing the expected return that is subject to given capacity and risk constraints.

Simulation-based optimization. A second approach is to use simulation-based optimization driven by historical data and learned models of competitor behaviour. First, machine learning can be used to learn patterns of how competitors have bid in the past and to construct conditional probability distributions for their future bids, conditioned on predicted market states such as demand or price forecasts. Given a parametric bidding rule, for example

$$p(\alpha) = c + \alpha(\hat{P} - c),$$

where c is our marginal cost, $\hat{P}$ is the forecasted market clearing price, and $\alpha \in [0, 1]$ is an aggressiveness parameter, Monte Carlo simulations can be run to generate a distribution of profits. For each α, we simulate competitor bids, run the market clearing process, and compute profits

$$\pi = q_{\text{disp}} \times (\text{MCP} - c),$$

where q_{disp} is the dispatched quantity and MCP is the market clearing price. From these simulations we obtain mean profit $\mu_\pi(\alpha)$ and standard deviation $\sigma_\pi(\alpha)$, and define a risk-adjusted utility

$$U(\alpha) = \mu_\pi(\alpha) - \kappa\, \sigma_\pi(\alpha),$$

where κ represents the decision maker's risk tolerance. By comparing $U(\alpha)$ across α, we identify bidding strategies that balance expected profit and risk. This framework naturally

extends to multi-block bidding, in which total capacity is split across several price–quantity blocks with different aggressiveness parameters, providing a more flexible offer curve that better matches real-world bidding.

A brief review of reinforcement learning algorithms. Reinforcement learning (RL) methods can be broadly grouped into three classes: there are value-based methods, policy-based methods, and actor-critic methods. Value-based methods, such as Q-learning, aim to estimate the long-run value of taking a given action in a given state and then choose actions that maximize this value. These methods are normally effective when the action space is discrete and reasonably small. Policy-based methods instead optimize the policy directly, typically by adjusting parameters to increase expected cumulative reward. They are especially useful when actions are continuous or when stochastic policies are desirable. Actor-critic methods combine both ideas by maintaining one component that evaluates actions (the critic) and another that updates the policy (the actor), often leading to improved stability in more complex environments[22]. In recent years, deep reinforcement learning methods such as Deep Q-Networks (DQN), Proximal Policy Optimization (PPO), and Deep Deterministic Policy Gradient (DDPG) have extended these ideas to high-dimensional settings[16].

MDP and Q-learning-based reinforcement learning. The third design, and ultimately the chosen design solution is to formulate the bidding problem as a Markov Decision Process (MDP) and solve it using Q-learning. The MDP framework provides a principled mathematical structure for sequential decision-making under uncertainty, naturally capturing the key elements of the bidding problem: the market state, the set of admissible bids, the profit-based reward, and the transition dynamics induced by the market. Q-learning can be thought of as an agentic approach to RL, where the agent focuses on making the next move that will maximize overall value. It maintains a quality score $Q(s, a)$ for each state–action pair and updates these scores using the Bellman equation. The temporal-difference update equation for Q-learning is

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha\Big(R_{t+1} + \gamma \max_{a'} Q(s_{t+1}, a') - Q(s_t, a_t)\Big).$$

As the agent is run through many episodes, it updates the Q-table based on observed rewards and gradually learns which actions are most beneficial in each state. Importantly, Q-learning does not require an explicit model of the market transition dynamics. This makes it suitable for electricity markets where rich historical datasets are available but the full market model is complex or unknown. This approach for a single agent is easily generalizable to multiple agents who bid at once on the same market, all learning how to optimize their own profit through Q-learning.

2 Model

The formal development of the mathematical model that supports the bidding optimization strategy will be done in this chapter, together with the MDP formulation, and each of the agents respective objectives.

2.1 Decision Makers

We model this bidding problem from the view of the decision makers, with each one of these decision makers acting as a natural gas generator that would participate in an ISO day-ahead market. All of the decision makers observe the same state and bid into the same market. When it comes to the objective, each decision maker wishes to maximize its own long-run profit.

All of the opponents are sorted into a supply stack, which is organized based on fuel type. Each opponent has their bid generated from this stack at each time step. This way our agents face a data-driven representation of competition.

2.2 Markov Decision Processes (MDP)

To formalize the sequential nature of bidding decisions, we cast the problem as a Markov Decision Process (MDP). An MDP is defined by a tuple

$$\mathcal{M} = (\mathcal{S}, \mathcal{A}, \mathcal{T}, R, \gamma),$$

where:

- $\mathcal{S}$ is the state space, describing the relevant market conditions known at the time of bidding.
- $\mathcal{A}$ is the action space, representing the set of admissible bids the decision maker can submit.
- $P(s' \mid s, a)$ is the transition kernel, giving the probability of moving from state s to state s' when action a is taken.
- $R(s, a, s')$ is the reward function, capturing realized profit (and any penalties) when transitioning from s to s' under action a.
- $\gamma \in (0, 1)$ is the discount factor, weighting future rewards relative to immediate ones.

At each episode $t = 0, 1, 2, \ldots$, the decision maker observes a state $s_t \in \mathcal{S}$, chooses an action $a_t \in \mathcal{A}$, receives a random reward $r_{t+1} \in \mathbb{R}$, and transitions to a new state s_{t+1} according to the transition kernel. A policy π specifies how the decision maker selects actions as a function of the current state.

2.3 Designing the MDP Model

The core of this project is defining the state space $\mathcal{S}$ and action space $\mathcal{A}$ on which the Q-learning algorithm will act. Formally, an optimal bidding policy $\pi^* : \mathcal{S} \to \mathcal{A}$ maps market conditions to a bidding decision such that long-run expected profit is maximized.

The state $s_t \in \mathcal{S}$ is constructed from historical load and LMP data, together with a load forecast and a time-of-day feature. Concretely, the state at each hour t is defined by the following variables:

- $L_t^{\text{hist}} \in \mathcal{L}^k$: the total system load in MWh, discretized into k bins.
- $\lambda_t^{\text{hist}} \in \Lambda^k$: the day-ahead locational marginal price (LMP) in \$/MWh, discretized into k bins.
- $\hat{L}_t^{\text{DA}} \in \hat{\mathcal{L}}^k$: a 14-day horizon load forecast, discretized into k bins.
- $\tau_t \in \mathcal{H}$: a categorical time-of-day feature that partitions each hour into one of five periods: night, morning, afternoon, evening, or late night.

The resulting state space is the Cartesian product

$$\mathcal{S} = \mathcal{L}^k \times \Lambda^k \times \hat{\mathcal{L}}^k \times \mathcal{H},$$

which is finite, as required for tabular Q-learning. The state vector is then $s_t = (L_t^{\text{hist}}, \lambda_t^{\text{hist}}, \hat{L}_t^{\text{DA}}, \tau_t) \in \mathcal{S}$.

The action space $\mathcal{A}$ is defined in terms of the bids the decision maker can submit. Let $\mathcal{P} = \{p_0, p_1, \ldots, p_{N_P-1}\}$ be a finite set of discretized bid prices in \$/MWh, and let $\mathcal{Q} = \{q_0, q_1, \ldots, q_{N_Q-1}\}$ be a finite set of discretized bid quantities in MW, ranging from 0 to a maximum capacity $q_{\max}$. Each action $a_t \in \mathcal{A}$ corresponds to a price-quantity pair $a_t = (p_i, q_j)$, and the full action space is the Cartesian product

$$\mathcal{A} = \mathcal{P} \times \mathcal{Q},$$

so that $|\mathcal{A}| = N_P \times N_Q < \infty$.

2.4 Reward / Cost Function

Potential users of this model, such as asset owners or analytics firms, strategically bid to achieve high long-run profitability while avoiding extreme risk. In the implemented environment, the per-step reward for each RL agent is defined directly in terms of profit and an explicit penalty for violating a notional risk constraint.

For agent i at a given delivery hour, let q_i denote the quantity of energy that clears from its bid, and let P_{clear} be the uniform market clearing price produced by the supply stack model. Each agent is also assigned a marginal cost c_i, sampled from a Henry Hub natural gas price

distribution (converted to \$/MWh) or, in a fallback case, inferred from historical LMPs. The realized profit for that hour is

$$\pi_i = q_i\big(P_{\text{clear}} - c_i\big).$$

To impose a simple risk constraint, the framework defines a maximum notional exposure based on a high quantile of historical LMPs. If an agent's chosen action would exceed this limit, the action is projected back into the feasible set and a risk penalty is applied. Specifically, if $\text{Notional}_{\text{orig}}$ is the original (price $\times$ quantity) exposure and $\text{Notional}_{\max}$ is the allowed maximum, a penalty of the form

$$\text{Penalty}_i = \lambda\big(1 + \text{severity}_i\big), \qquad \text{severity}_i = \max\left\{0,\ \frac{\text{Notional}_{\text{orig}} - \text{Notional}_{\max}}{\text{Notional}_{\max}}\right\},$$

is subtracted from the profit whenever a clip occurs, where $\lambda \geq 0$ is a tunable risk penalty parameter. The per-step reward used by the RL algorithm is then

$$R_i = \pi_i - \text{Penalty}_i.$$

The reward system clearly demonstrates the tension between aggressive bidding (for larger volumes and profits at high prices) and sticking to the risk budget constraint. The reward system adheres to the general principle of the CAPM approach that high returns entail high risks, but yet it is simple enough for Q-learning tables.

2.5 Transition Kernel

As mentioned before, the complete MDP formulation involves transition kernels $\mathcal{T}(s' \mid s, a)$. For electricity markets, such transitions depend on many intricate and evolving variables, including future demand, production from renewable energy sources, prices of fuels, grid limitations, as well as other companies' bids. Formulating the transition kernel explicitly is hard and requires strong modeling assumptions.

A model-free and empirical approach is more natural for Q-learning applications, and the data can be obtained based on historical ERCOT load/LMP time series, ERCOT load forecasts, and a supply stack using generators. From the agents' perspective, this induces an implicit transition kernel $\mathcal{T}(s' \mid s, a)$ that is not modeled analytically but is instead sampled through the above simulation procedure. Q-learning only requires samples of transitions and rewards, not a closed-form expression for $\mathcal{T}$, making this approach appropriate for the problem at hand.

2.6 Decision Maker Objective

The decision maker's objective is to select a policy π that maximizes long-run expected cumulative reward. Formally, starting from an initial state s_0, the value of a policy π for a given agent is

$$V^{\pi}(s_0) = \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t R_{t+1} \,\middle|\, s_0\right],$$

where the expectation is taken over the randomness in market data, opponent bids, and any stochastic elements of the policy. The goal is to find an optimal policy π^* satisfying

$$\pi^* \in \arg\max_{\pi} V^{\pi}(s_0),$$

for relevant initial states s_0.

In the context of this thesis, this objective corresponds to learning bidding strategies that:

- Achieve high expected profit over many market days;
- Respect an empirically calibrated notional risk limit through the clipping and penalty mechanism;
- Remain robust to stochastic market conditions and a realistic, fuel-based supply stack of competing generators.

The tabular Q-learning algorithm is used to approximate π^* by learning state–action values $Q(s, a)$ and choosing actions that are greedy (or nearly greedy) with respect to these values.

2.7 Existence of Optimal Policies

Under standard assumptions, finite-state, finite-action MDPs admit optimal stationary policies. In particular, when $\mathcal{S}$ and $\mathcal{A}$ are finite and the discount factor satisfies $\gamma \in (0, 1)$, there exists at least one stationary policy π^* that maximizes the value function $V^{\pi}(s)$ for all states s. Furthermore, the optimal value function satisfies the Bellman optimality equation

$$V^*(s) = \max_{a \in \mathcal{A}} \mathbb{E}\big[R_{t+1} + \gamma V^*(s_{t+1}) \mid s_t = s, a_t = a\big],$$

and the corresponding optimal Q-function satisfies

$$Q^*(s, a) = \mathbb{E}\big[R_{t+1} + \gamma \max_{a'} Q^*(s_{t+1}, a') \mid s_t = s, a_t = a\big].$$

In the multi-agent setting considered in this thesis, each agent i maintains its own Q-table $Q_i(s, a)$ and independently applies the same convergence results, treating the other agents' policies as part of the environment. The optimal value function for agent i satisfies

$$V_i^*(s) = \max_{a \in \mathcal{A}} \mathbb{E}\big[R_{i,t+1} + \gamma V_i^*(s_{t+1}) \mid s_t = s, a_{i,t} = a\big],$$

and the corresponding optimal Q-function for agent i satisfies

$$Q_i^*(s, a) = \mathbb{E}\big[R_{i,t+1} + \gamma \max_{a'} Q_i^*(s_{t+1}, a') \mid s_t = s, a_{i,t} = a\big].$$

For an MDP with finite state and action spaces, if each state–action pair is visited infinitely often and the learning rate is chosen to satisfy standard diminishing-step-size conditions, Q-learning converges to the optimal Q-function with probability one[30]. Consequently, the greedy policy with respect to the learned Q-table converges to an optimal stationary policy. By discretizing historical load, LMP, forecasted load, and time-of-day into finitely many bins, we obtain a finite MDP in which these existence and convergence results apply, providing theoretical justification for the use of tabular Q-learning in this thesis.

3 Learning Algorithm

As mentioned in section 2, the identified stochastic game is an MDP. The following section explores the algorithm used by the multi-agent network to achieve equilibrium in said game. The focus is on the underlying architecture for the implemented Q-learning algorithm (i.e., the specific temporal difference equation used), the architecture for simulation opponent agents, and implementation strategies to ensure reproducible experimentations (and thus trustworthy results).

3.1 Q-Learning

As previously explained (in greater detail), the discretizations of both the state and the action space allow for the use of tabular Q-learning (as opposed to using neural network approximation, as would be required in a continuous environment)[22]. The generic tabular Q-learning update formula is

$$Q_{t+1}(x_t, u_t) = Q_t(x_t, u_t) + \alpha_t(x_t, u_t)[r_t + \gamma \max_{u'} Q_t(x_{t+1}, u') - Q_t(x_t, u_t)]$$

where $Q_t(x, u)$ is the current estimate of a state-action pair at time t, α_t is the learning rate at time t, r_t is the one stage reward for taking action u_t in state x_t, γ is the discount factor and $\max_{u'} Q_t(x_{t+1}, u')$ is the current estimate of the best next stage value [31]. Modifying this for the multi-agent case, we get a modified temporal difference of

$$Q_{i,t+1}(x_{i,t}, u_{i,t}) = Q_{i,t}(x_{i,t}, u_{i,t}) + \alpha_{i,t}(x_{i,t}, u_{i,t})[r_{i,t} + \gamma \max_{u'} Q_{i,t}(x_{i,t+1}, u') - Q_t(x_{i,t}, u_{i,t})]$$

where each individual agent is indexed by i. What is key to note here is that each agent also has their own independent learning rate, meaning agents do not and need not necessarily learn at the same rate, as discussed in greater detail below[15]. Finally, explicitly plugging in the reward function discussed in section 2.4, we arrive at our solution's update equation:

$$Q_{i,t+1}(x_t, u_{i,t}) = Q_{i,t}(x_t, u_{i,t}) + \alpha_{i,t}\left[(p_t^* - \hat{c}_{i,e})q_{i,t}^{\text{clear}} + \gamma \max_{u_i'} Q_i(x_{t+1}, u_i' - Q_i(x_t, u_{i,t}))\right]$$

As per how agents select actions during their training phase, agents follow Boltzmann exploration (also known as Softmax exploration). This exploratory policy works by converting the Q-table into a probability distribution, placing higher probability on higher Q-values and vice-versa through the following formula:

$$\pi_t(u \mid x) = \frac{e^{Q_t(x,u)/\tau_t}}{\sum_{v \in \mathcal{U}(x)} e^{Q_t(x,v)/\tau_t}}$$

In this formula, $\tau_t > 0$, the temperature parameter, controls how exploratory the agents behavior is. For very large τ_t, the distribution is near uniform, and as $\tau \to 0$, the exploratory policy becomes near greedy.

Regardless of how small an action's associated Q-value (for a given state) may be though,

there is always a positive probability that it will be selected. This results in improved value estimations comparative to just a pure-greedy policy, as there is always a chance for the model to self correct [26].

By default (in our solution, that is), τ_t is updated according to a Q-gap function of the delta between the largest and second largest Q-values (for a given state). The output of this function is a large τ (more exploratory) when the delta is small, and a small τ when the delta is large.

The choice of exploratory policy and update rule (for τ) is non-trivial considering the high variability of market clearing prices and opponent bids in the game. Boltzmann finds the proper balance between exploration and greediness, while the Q-gap method allows the agent to confidently develop a policy while properly learning the market.

Although each agent's learning rate $\alpha_{i,0}$ is initialized to the same value, they are updated according to the frequency of state-action pairs that an agent realizes:

$$\alpha_{i,t}(x,u) = \frac{\alpha_{i,0}}{N_{i,t}(x,u)^{\xi}}$$

The learning rate thus diminishes over time, by a magnitude directly correlated by the experience each agent has with said state-action pair to an exponent $\xi \in (0.5, 1]$. Consequently, actions do not share a step-size, even within the same state, or within the same agent. Although this results in large ΔQ's at the start of training, over enough episodes, this causes $\Delta Q \to 0$, helping with convergence to a stable policy. This is especially relevant in the context of an ISO market, which often has random outlier market clearing prices (which would alter late-stage training significantly, were the learning rates constant) [28].

3.2 Market Simulations

The historical locational marginal price (herein referred to as LMP) within ERCOT is used as an anchor for the outcomes of bids within the simulator. Similarly, historical load data is used as proxy for market demand. The simulator also takes advantage of fuel mix data (i.e., the historical percentage breakdown of load by fuel type) as well as a dataset containing the active generators within ERCOT (also filtered by fuel type). This data is used to generate heterogeneous opponent bids for the Q-agents to compete against, thus properly reflecting the intended nature of the game, that is, to optimally compete within an ISO, rather than learn how to undercut historical LMP. The simulator also takes advantage of a historical Henry Hub natural gas prices data, which is a distribution, giving the solution a realistic proxy to sample marginal costs from. Rather than learning how to play around a fixed marginal cost, this forces agents to also factor in the unpredictability in marginal cost faced by real life gas generators, and thus intrinsically, the opponent bids within the simulator.

The simulator is also responsible for converting the continuous market data into the state space model laid out in section 2.2 so that tabular Q-learning is feasible. To achieve this, historical and forecasted load alongside historical LMP is discretized into a number of bins.

The plots and data generated within this paper have used 8 bins. These discretizations occur before every single episode. These bins are then joined with a time of day feature to produce the observations seen by the agents. Rather than treating each hour as its own category, the agents observe the time as morning, afternoon, evening, night, or late-night.

Episode construction also occurs by the simulator. The simulator uses a 14 day rolling window, with each day further broken up into 24 hours. Consequently, agents make 336 (14×24) observations per episode. Each episode, the state is shifted by 24 hours. There is therefore a 13 day overlap in the rolling window.

It should also be noted that the discretization method is dynamic. The four candidate discretization methods are uniform, quantile, sigma-based and hybrid binning. The model's discretizer prioritizes the method that results in the least amount of empty bins, and then subsequently picks the most balanced bin of the remaining candidates.

The simulator is also responsible for implementing the action space defined in 2.2. The simulator represents bids as a (price, quantity), which also fit into some discretized bin. The prices bins are generated using historical LMP data to ensure a realistic range. As per quantity, the bins are fixed within a range of 0 MW to an amount reflective of the average max capacity of a generator within ERCOT. This is to ensure that agents have a realistic impact on the market with their bids (i.e. they are unable to service the entire market at some nominal price). The discretization of the action space results in a feasible action space for tabular Q-learning that doesn't lose out on the intricacies of real ISO bidding.

Once bins have been defined for both price and quantity, the simulator interprets their combination as cartesian products, i.e., each pair is viewed as a single action. In essence, the simulator is decoding an action into a price and quantity. The encoding of price and quantity converts a naturally continuous environment into something Q-learning feasible without losing the economic meaning of the action, ensuring agents are learning the intended game.

One of the core features of our solution is that instead of bidding against just the historical LMP, agents are competing at every time step against opponents who's behavior is proxied from real data. Using a dataset of all generators that were active in ERCOT over the given training period, opponent agents are generated with a fuel type and load capacity. These opponent agents are then grouped by their fuel type, such that the simulator doesn't treat all agents as homogeneous bidders, but rather bidders following a unique (to the fuel type) set of behaviors. The simulator then uses a historical fuel mix (indexed by the hour) to segment load by said fuel type.

Relative to the capacity of each opponent, a quantity of each historical fuel segment is assigned to each opponent. Consider the following trivial example. Suppose generator A has a capacity of twice the capacity of generator B, whom are the sole generators of their fuel segment. Then if there is 6 MW of capacity available at a given time t, 4 of that is assigned to A and 2 is assigned to B. This is of course a simplification of what historically occurred at time t, as it is unknown what these generators actually bid, but this still represents a fair

representation for what likely roughly occurred.

As per the price that each opponent agent bids, the simulator first orders the fuel stack by a predefined stack ordering, based on average prices by fuel price. Then within each segment, a mean zero variance one Gaussian distribution is generated relative to where that segment sits on the stack. Suppose there is a segment taking the top 10% of the stack (i.e. the highest priced bids). Also suppose that the LMP at this given time t is \$50. Then a Gaussian distribution between $(1 - 0.1)(\$50) = \$45$ and $(1 - 0.0)(\$50) = \$50$ is generated. Each opponent agent's with a corresponding fuel type is then sampled from this distribution. The resultant opponent bid is thus neither deterministic nor arbitrary.

Once the simulator has access to both the opponent bids and the RL bids, both are combined into a single stack. Similar to the real market, the simulator's stack is merit-order clearing, meaning it is possible for both the opponent and RL agent's bids to be rejected. The last accepted bid sets the market clearing price rather than using the historical LMP. This pipeline is what allows are agents to truly learn how to compete, both against themselves and non-RL competitors.

Using a simulator presents two major design advantages. Firstly, it allows agents to be trained in a controllable environment, where key policy-effecting variables can manually be perturbed. This in turns allows for ablation studies, meaning that we can identify what a certain policy perform better than another. Secondly, it also allows for testing on how sensitive policies are to the various assumptions we have had to make in the process of making the solution feasible to implement. Several of these assumptions are non-trivial, and thus it would be irresponsible to not verify whether our agents are truly learning the intended game.

3.3 Baseline Bidding Strategies

Considering that the solution is also being produced for quasi-commercial purposes, policy convergence is not enough for "meaningful" learning. There have also been economic baselines, or "unintelligent" agents, created to compare how the agents' perform comparative to non-RL strategies. While this section focuses on the technical details of these baselines, section 4.4 contains a comparison of performance and the implications of those results.

The first baseline strategy is a rule-based policy, cost-plus markup. The price of each bid is set to the marginal cost of the generator plus a constant percentage markup markup, i.e.

$$\text{price}_t = (1 + \text{markup}) \times \hat{c}_t$$

The quantity is fixed no matter what at a certain output. The resultant (price, quantity) pair is then discretized by the simulator, making the comparison to our Q-learning agents more fair. A baseline anchored by marginal cost is a quality baseline for this project, considering marginal cost drives much of the current bidding prices in real life ISOs[9]. It is a pretty safe and trivial assumption that this strategy will never be optimal, hence why it is also a quality engineering heuristic. It does not make sense to implement a much more complicated learning framework that cannot beat such a simple strategy.

The other baseline strategy, a historical-quantile strategy, is more data-driven. Bidding prices are instead derived from historical LMPs, meaning it is possible (although unlikely) for this strategy to lose money, similar to the RL agents. Once the historical quantiles have been computed for the agent though, they remain static, hence the lack of learning. Similar to the cost-plus markup strategy, the quantity portion of each bid is kept fixed. The price component however is chosen from an LMP constructed lookup table, which sorts prices based on the time of day (see section 3.2 for a time of day explanation). The resultant $(\text{price}, \text{quantity})$ action is then encoded by the simulator into a discrete value compatible with the action space. The historical quantile baseline is a good proxy for the RL agents, as it is driven by empirical price information without any policy revision. From an engineering perspective, it allows us to verify that our RL agents aren't performing better than a non-learning strategy simply because of the data we are feeding it.

3.4 Comparison of Algorithm Choices

The biggest design decision in this project was to use tabular Q-learning. Since both the state space and action space were discretized, using tabular reinforcement learning as opposed to neural network approximation became feasible. Moreover, we were heavily encouraged by our supervisor to consider tabular Q-learning for its transparency, especially with how every state-action pair can be directly examined.

The final implementation also includes several smaller design decisions regarding benchmark baselines, exploratory policies, warm-starting and action feasibility restrictions. Some of these became part of the default configuration for the model, while others were eventually deprecated. The following table aims to give a rationale for those which were retained, which are available as configurable runtime options within the model.

Method	Role	Main Pro	Main Con
Tabular Q-Learning	Policy Training	Interpretable results	Requires smaller, discretized state/action space
Cost-plus markup	Baseline	Intuitive benchmark	Static rule, no improvement
Historical quantile	Baseline	Makes use of historical data	No learning, does not adapt to reward
Fixed τ Boltzmann	Exploratory Policy	Simple and stable	Too much explorations in late stage training
Decaying τ Boltzmann	Exploratory Policy	Exploratory early, greedy later	Decay schedule is manually chosen, introducing bias
ΔQ Boltzmann	Exploratory Policy	Exploratory nature of agent is anchored by confidence Q-table	Depends on quality of ΔQ data. Perhaps misleading.
Warm-Start	Initialization	Improves efficiency of early learning	Introduces a new source of bias
Feasibility constraints	Operational Rule	Prevents overly aggressive or unrealistic bids	Can alter final encoded action relative to raw decoded action

What should be noted from the above table is that there is only one main framework for generating policies. There are two baselines such that they can be compared against each other as well.

The implemented solution reflects a waterfall of design choices. The discretization of the state and action space made it such that tabular Q-learning could be implemented. Rule based baselines consequently make for good benchmarks to see the added benefit of an adaptive algorithm. The other design choices aid in widening gap between the Q-learning agents and these baselines.

3.5 Engineering Implications of the Learning Framework

The biggest tradeoff from an engineering perspective within our model is tractability versus realism. In order to make RL feasible within the constraints of this project, several aspects of the market had to be simplified within the model's simulator. However, if the market is made too simple, there is no longer a point in training on it. If the market is aggressively changed, the agents will end up learning a different game than intended.

One major example of this tradeoff is how the simulator discretizes the state and action space. Variables like time, price and load are continuous in real life, but need to encoded into discrete bins for the purposes of tabular Q-learning. This is a necessary engineering decision to make Q-learning even possible, but the discretization of this information is not lossless due to the binning.

The model also greatly simplifies opponent bidding behavior. Although the model's simulator uses real data as a proxy for opponent bids, they are still just samples from a Gaussian distribution, and all opponents submit a bid at every time step. From an engineering perspective, this is a reasonable compromise, as the market is not being treated as an exogenous price and rather independent competitors. Consequently, the key parts of the game we wish the RL agents to learn are still being captured by the simulator.

The core theme from our design decisions is that the model is not trying to maximize its realism, but rather focus on retaining key parts of market behavior needed for our RL agents to learn the intended game. A richer simulator would likely need a richer learning algorithm alongside it, such as a DQN (neural network approximation), which is out of the scope for this project.

Another engineering implication is the restriction on allowable actions by the RL agents. Before being submitted to the market, the simulator analyzes bids, enforcing pricing and quantity limits. As a result, the RL agents are maximizing their rewards subject to these bounds. This makes sense in the context of the real life market though, as the policy isn't useful if it isn't compatible with physical and financial limitations of real life generators. This projection of actions onto a bounded subset ensures that bids remain operationally credible.

It should also be noted that the credibility of an RL agent's results are a reflection of the credibility of its environment. For example, poor market clearing logic results in poor reward signals. The simulator itself is the game the agent is learning, not the true market.

3.6 Design Procedure

The model adheres to the following process:

1. Historical LMP and load data are loaded and clipped to a given date range.
2. Forecasted load data is converted to UTC, and aligned with historical load data.
3. Discretizers are fit for the state space variables, and then assembled in a window object (i.e., 14 day window)
4. The price component of the action space is constructed by discretizing bid price bins using historical LMP data as a proxy.
5. Similarly, the quantity component of the action space is constructed by constructing bins starting from 0 ranging to a fixed constant.

6. The price and quantity bins are flattened into a 1-dimensional space.
7. Active (historically) generator data is used to create an equivalent (historically) amount of opponent agents.
8. Henry Hub historical price data is converted into a marginal cost distribution, which is sampled by RL agents during training.
9. Episodes are generated over a 14 day window. At each timestamp, agents observe the state space, and subsequently place a bid using the Boltzmann exploratory policy.
10. The selected action is project into a subset of feasible actions. The vast majority of the time, this is the same action as before.
11. Opponent bids are sampled from fuel-band price ranges (Gaussian distribution).
12. The RL bids are decoded from their discretized action space.
13. All bids (both RL and non-RL) are subjected to market clearing rules.
14. The resulting reward is used to update the agents' Q-tables.
15. Each agent's learning rate is harmonically adjusted.
16. Once training is complete, policies are saved and used to compete against baseline benchmarks.
17. The process may be repeated with varying control studies such as market elasticity, number of RL agents, etc.

4 Simulation Results

This section presents the simulation results obtained by applying the tabular Q-learning framework described in Section 3 to historical ERCOT electricity market data. The central aim is to provide training and testing simulation results. Simulation environment and data configuration, parameter selection as part of key design choices, training convergence behavior, and baseline performance on out of sample data are discussed, alongside design iterations and a sensitivity analysis using cumulative reward per agents.

4.1 Simulation setup

Market: The simulation was configured to run within the Electricity Reliability Council of Texas (ERCOT) market.
Training period: A training period from January 1st, 2025 to November 25th, 2025 was utilized. Note that November 25th, 2025 is the latest available data point at time of training. Given the episode construction, this resulted in a maximum of 314 available training episodes; runs were capped at 350 but did not exceed the available data window.
Data Sources: Given the construction provided in Section 3, data sources for forecasted

load, actual load, actual fuel mix, and actual locational marginal prices (LMPs) were utilized. Enverus' 2025 ERCOT load forecast was utilized for forecasted load. Enverus' ERCOT-wide LMP, representing the average LMP price across all nodes, was utilized for actual locational marginal prices [6]. Actual load and fuel mix from the ERCOT market data access portal was utilized [4]. As part of our iterated design, the ERCOT day-ahead demand forecast was utilized from the ERCOT market data access portal [4].
Evaluation period: Learned policies were evaluated against 2019-2024 market data as part of the baseline tests in section 4.4. Historical day-ahead forecasted demand was utilized from the ERCOT market data access portal [4].
Agent Number Configurations: Experiments were run with 1, 3, and 5 competing agents respectively.

4.2 Parameter Selection

As part of our design, the team decided to conduct a parameter tuning on model components such as learning, temperature, convergence threshold, and risk. A full list of parameters as is provided in Table 10.

Learning: Warm Start, policy freeze k, learning rate, and learning rate exponent.

When the `warm start q` flag is set to `True`, results in the following Q-table for agent i:

$$Q_{i,0}(x_i, u_i) = \bar{q}_{i,0}$$

where $\bar{q}_{i,0}$ is derived from the cost-plus markup baseline:

$$\bar{q}_{i,0} = Q_{\text{Cost Plus Markup, 0}}(x_i, u_i)$$

.
Table 1 demonstrates the maximum agent ΔQ during the last 5 training episodes for our agents with the `warm start q` flag set to `True` and `False` across 1, 3, and 5 agents.

Number of agents	Warm Start: True	Warm Start: False
1	$\approx 22 - 25$	$\approx 18 - 35$
3	$\approx 20 - 30$	$\approx 18 - 50$
5	$\approx 15 - 20$	$\approx 15 - 30$

Table 1: maximum agent ΔQ during the last 5 training episodes with and without warm start

As shown in the above table, warm starting the Q-table across single agent and multi agent setups leads to a smaller neighborhood of convergence with respect to our ΔQ values. Additionally, warm start helps make early stage exploration more guided. This is described with Table 2 below, with the single agent case showing significant reduction in the mean agent ΔQ. Warm start ablations were also produced to support our results, as shown in Figure 18 in the appendix.

Number of agents	Warm Start: True	Warm Start: False
1	233.85	331.57
3	373.04	318.29
5	393.11	352.89

Table 2: Mean maximum agent ΔQ during the first 25 training episodes with and without warm start

For our policy freeze K values in the multi-agent case, our team performed the ablation in Figure 1 with 2, 5, and 10 agents, for K values of 2, 5, and 10.

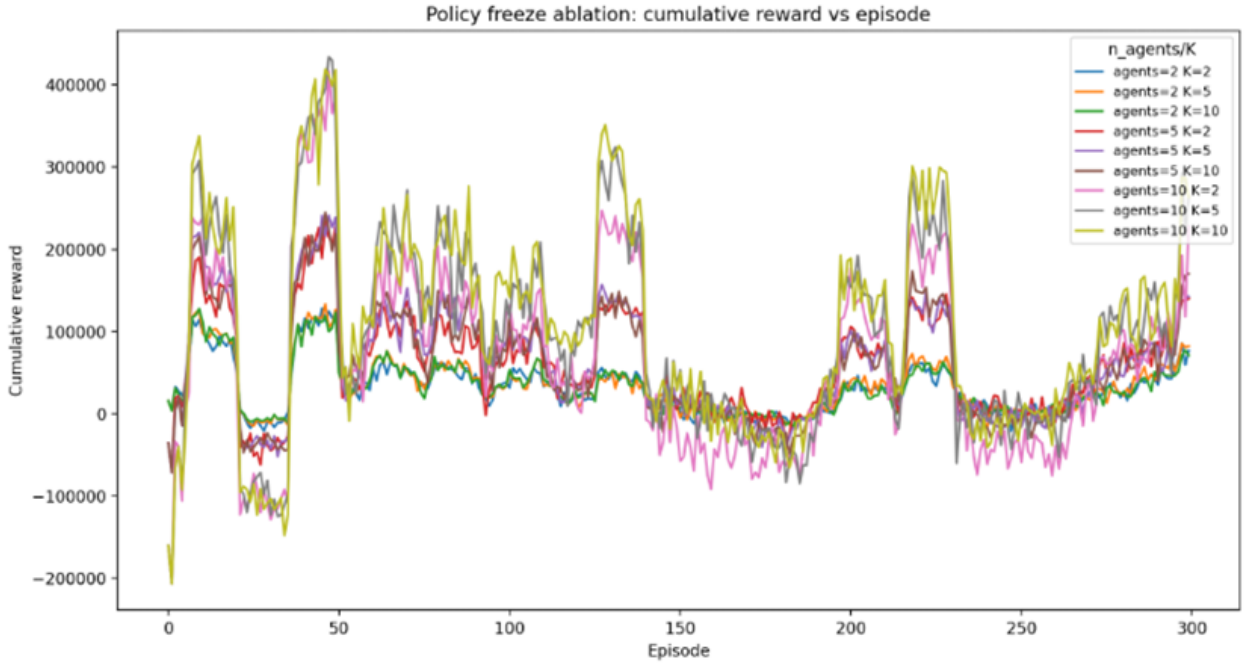


Figure 1: Policy Freeze K ablation in multi-agent setting

Given the relatively comparable performance across each policy freeze K value, a value of $K = 10$ was selected as it enables a longer uninterrupted learning period for each agent, providing more Q-table updates per freeze cycle before the policy is re-evaluated by other RL agents. In the single-agent case, a shorter interval of $K = 5$ was used, since there is no opponent non-stationarity to average over and the agent's own policy is the only source of environmental change.

The learning rate schedule used in Section 3.1 takes the generalized harmonic form

$$\alpha_{i,t}(x, u) = \alpha_{i,0}/N_{i,t}(x, u)^{\xi}$$

with $\alpha_{i,0} = 1.0$ and $\xi = 0.7$ fixed across all runs. A unit initial learning rate ensures that early Q-value estimates are accepted without dampening [22, 31], which is appropriate when the Q-table is uninitialized and large corrections from the first observed transitions are expected. The exponent $\xi = 0.7$ was chosen to satisfy the Robbins-Monro conditions as described in

6.1, while remaining below 1.0 to slow the rate of decay and preserve meaningful updates deep into training. The empirical effect of this schedule, a reduction of roughly three orders of magnitude in the effective step size over 350 episodes, is shown in 15.

Temperature: Value, Decay, and Minimum

The action entropy for agent i at training step t is computed as

$$H_i(t) = -\sum_{a \in \mathcal{A}} \hat{p}_i(a,t) \log \hat{p}_i(a,t), \tag{1}$$

where $\hat{p}_i(a,t)$ is the empirical frequency of action a observed in rollouts at step t, not the softmax output of the policy network.

As noted in Section 3.4, three temperature update rules were considered: fixed τ, decaying τ, and ΔQ Boltzmann. The ΔQ schedule drives temperature from the gap between the top-two Q-values, offering theoretically adaptive exploration. However, in a multi-agent setting, opponents are simultaneously learning, so Q-value gaps fluctuate not only from the agent's own uncertainty but from the shifting behavior of their opponents. This conflates opponent non-stationarity with value uncertainty, causing T to oscillate unpredictably and making the exploration schedule non-reproducible across runs with identical hyperparameters. Exponential decay was therefore selected.

The schedule is parameterized as

$$T_{\text{ep}} = \max(T_{\min},\ T_0 \times \text{decay}^{\text{ep}})$$

with $T_0 = 1.0$, decay $= 0.995$, and $T_{\min} = 0.1$. The choice of $T_0 = 1.0$ initializes the Boltzmann distribution at moderate diffuseness, which is sufficient to explore beyond the warm-start cost-plus initialization without wasting early episodes on random action selection. The decay factor of 0.995 spreads the exploration-to-exploitation transition smoothly across the 314-episode training window. A faster decay would commit the policy before Q-values have stabilized, while a slower decay would delay exploitation beyond the training data range noted in Section 4.1. The $T_{\min} = 0.1$ floor ensures a residual exploration probability is preserved through the late-stage BRGap evaluation window, preventing the policy from freezing entirely in the event of a market regime shift in late-year ERCOT data [28].

Early results suggested that non-constant schedules reduced action entropy faster and with less variance. The following figures demonstrate action entropy across a fixed and exponentially decaying temperature schedule in the single-agent case, with all other parameters held constant.

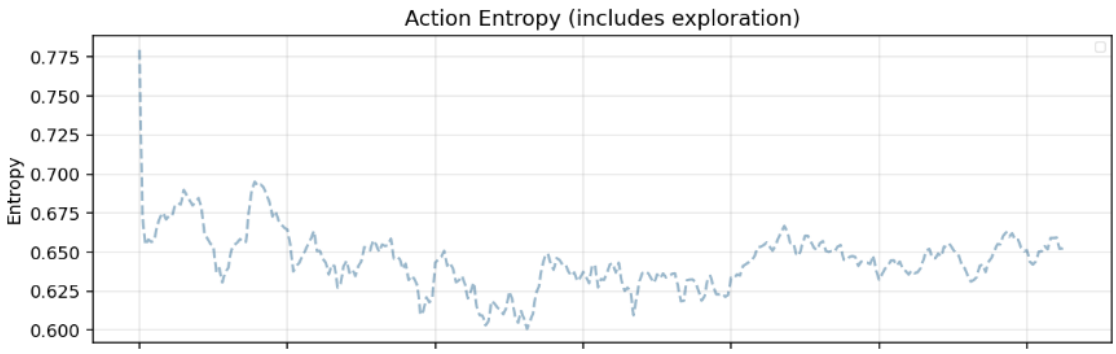


Figure 2: Action entropy across training period for single-agent case under exponential decay schedule.

Figure 3: Action entropy across training period for single-agent case under fixed temperature schedule.

Under a fixed temperature schedule, the exploration-exploitation balance is constant throughout training, causing early episodes to waste time exploiting a poorly initialized Q-table while late episodes continue exploring after the policy has converged. The exponential decay schedule in Figure 2 produces a lower entropy with a narrower range of values relative to the fixed schedule in Figure 3, confirming the decision to use a non-constant exploration schedule across all agent configurations.

Threshold: BRGap late save threshold, k eval, retrain episodes, late save k, and late save min episodes

The Best-Response Gap (BRGap) for agent i is defined as

$$\text{BRGap}_i = \max_{\pi_i} J_i(\pi_i, \pi^*_{-i}) - J_i(\pi^*_i, \pi^*_{-i}), \tag{2}$$

where $J_i(\pi_i, \pi^*_{-i})$ denotes agent i's average episodic reward when playing π_i against frozen opponents π^*_{-i} [12]. More intuitively, BRGap_i measures the maximum gain agent i could achieve by deviating unilaterally from the learned joint policy $\pi^* = (\pi^*_1, \ldots, \pi^*_N)$. Since computing a true best response is generally intractable, we approximate it empirically. For each agent i, a fresh best-response agent is trained against the frozen opponents π^*_{-i} over a

randomly sampled set of K episodes drawn from the training period, where K is defined in the parameters section. The approximated best-response policy is then

$$\hat{\pi}_i \approx \arg\max_{\pi_i} J_i(\pi_i, \pi^*_{-i}), \tag{3}$$

The Normalized Best-Response Gap is then defined as:

$$\text{BRGap}_{norm} = 1/n \sum_{i=1}^{N} \frac{\text{BRGap}_i}{J_i(\pi^*_i, \pi^*_{-i})} \tag{4}$$

BRGap_{norm} was utilized as a trigger for late policy saving. The threshold of $\varepsilon = 0.25$ defines a normalized ε-Nash equilibrium criterion, meaning a joint policy π^* is accepted if no agent can improve its reward by more than 25% through unilateral deviation [2], which is grounded in the framework discussed in Section 5.2. The 0.25 threshold ties this convergence guarantee to an empirical number. Rather than requiring exact Nash equilibria, it identifies policy snapshots where the incentive to deviate is bounded to a practically negligible fraction of current earnings. The specific value reflects a balance between meaningful Nash quality and robustness to estimation noise inherent in retraining over $K = 28$ episodes, representing $\approx 9\%$ of the available training window.

This sample size was constrained by two practical factors. The first is the computational cost of retraining a fresh best-response agent at each evaluation checkpoint, which scales with the number of agents, with full training runs requiring upwards of 3 hours. The second is the need to preserve sufficient held-out episodes for out-of-sample policy evaluation in Section 4.4, given the 314-episode training cycle.

The remaining threshold parameters govern when and how the late-save mechanism activates. The minimum episode floor of 230 ensures the Q-table has undergone sufficient training before BRGap evaluation is triggered.

Whether the threshold is satisfied is an empirical outcome reported in Sections 4.3 and 4.4. In the multi-agent case it was not triggered during training. The design improvements in Section 4.7 address these concerns.

Risk: `max notional q` and risk penalty λ

Two risk parameters govern the feasibility constraint applied to agent bids before market submission. The `max notional q` parameter sets the 95th percentile of the absolute historical LMP distribution as an upper bound on the notional value of any submitted bid $u_{i,t}$. This ensures that agents cannot submit bids that are economically unrealistic relative to the observed market [9]. As noted in Section 3.5, this projection onto a feasible action subset preserves operational credibility without materially restricting the learned strategy, since the vast majority of actions already satisfy the bound.

The risk penalty $\lambda = 0$ disables an optional reward penalty applied to clipped bids, whose original notional bid value exceeded max notional q before projection. A $\lambda > 0$ would subtract a severity-scaled penalty from the agent's reward at each clipped step, discouraging aggressive bidding. This was disabled across all runs as the observed clip rate was negligible throughout training and evaluation. The per-step clip rate was negligible across all configurations: mean 0.01% (peak 3.9%) for one agent, mean 0.03% (peak 8.1%) for three agents, and 0.0% for five agents. Bids were therefore already falling within the feasible region without penalization, making $\lambda > 0$ redundant and introducing an unnecessary reward shaping signal that could bias the learned policy away from the true profit objective.

4.3 Training results and Algorithm Convergence

With the simulation setup in Section 4.1 and the parameter configuration as described in Section 4.2, the following results were generated in both the single agent and multi-agent (3, 5 agent) case.

Single Agent Reward and Variance

Within the single agent case, the agent has highly variable total reward within the first 50 episodes of between $-$2700 and $66000. This is supported by the reward variance, which average's $338 within the first 50 episodes.

During the last 50 episodes, at tighter total reward of between $23000 and $70000 is observed. The reward variance also decreased to $150 per step.

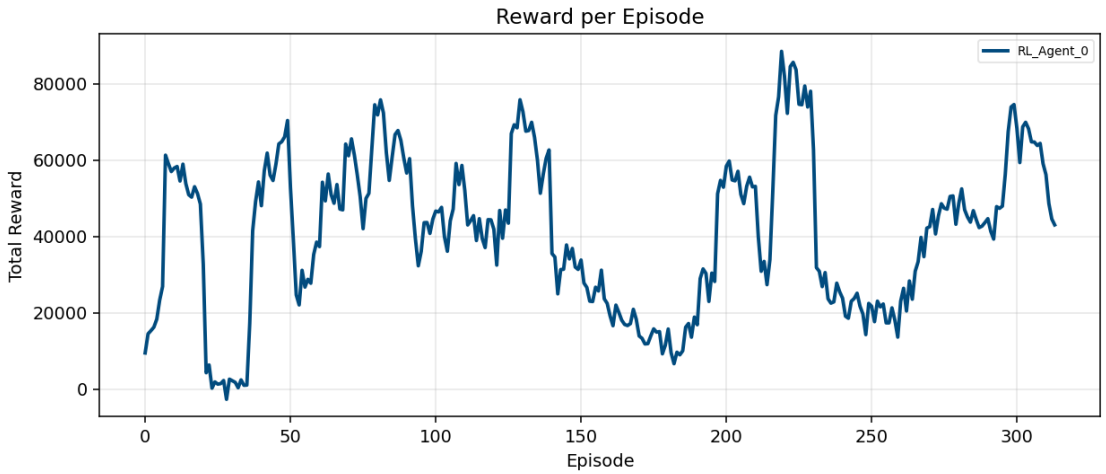


Figure 4: Training reward curve in single agent Case.

Figure 5: Rolling Reward variance for single agent case.

Given our reward calculation from Section 2.4, taking the variance of the clearing price can help normalize the mean and variance reward per episode. The first 50 episodes yielded an average variance of 68 per episode, while the final 50 episodes yielded an average variance of 73. Considering the ratio between the reward and clearing price ratio, it increased from 4.97 to 6.06.

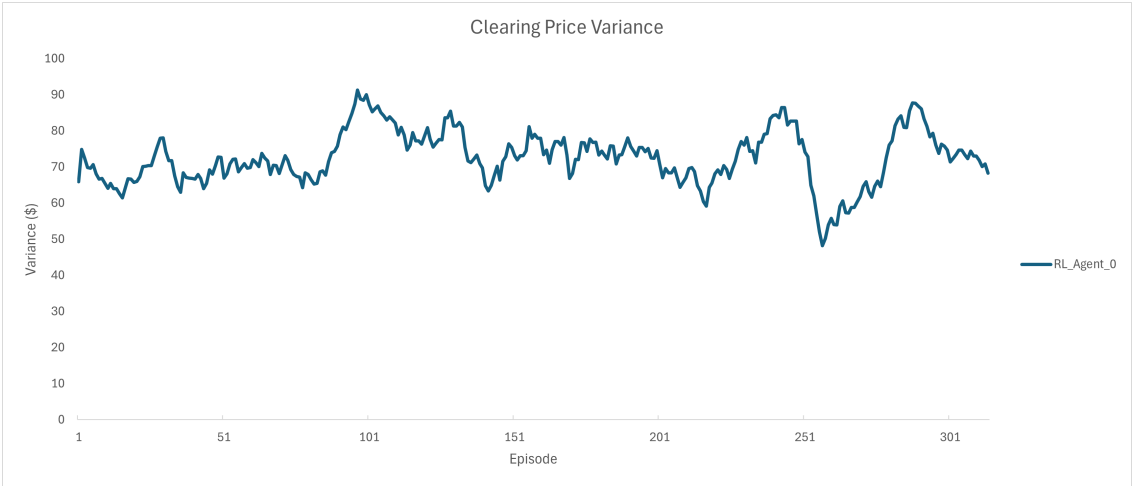


Figure 6: Clearing price variance across training episodes in single-agent case.

Figure 7: Ratio of reward to learning price variance across training episodes in single-agent case.

Convergence Criteria and Results

The ΔQ, or max inter-episode Q-table change, for agent i at training step t, defined as

$$\Delta Q_{i,t+1} = \max_{s,a} |Q_{i,t+1}(s,a) - Q_{i,t}(s,a)|, \tag{5}$$

The ΔQ measures Q-table self-consistency. As the Q-table converges to an optimal value function, successive Bellman updates produce diminishing corrections. A declining trajectory indicates that the Q-table is self-consistent with respect to the Bellman equation.

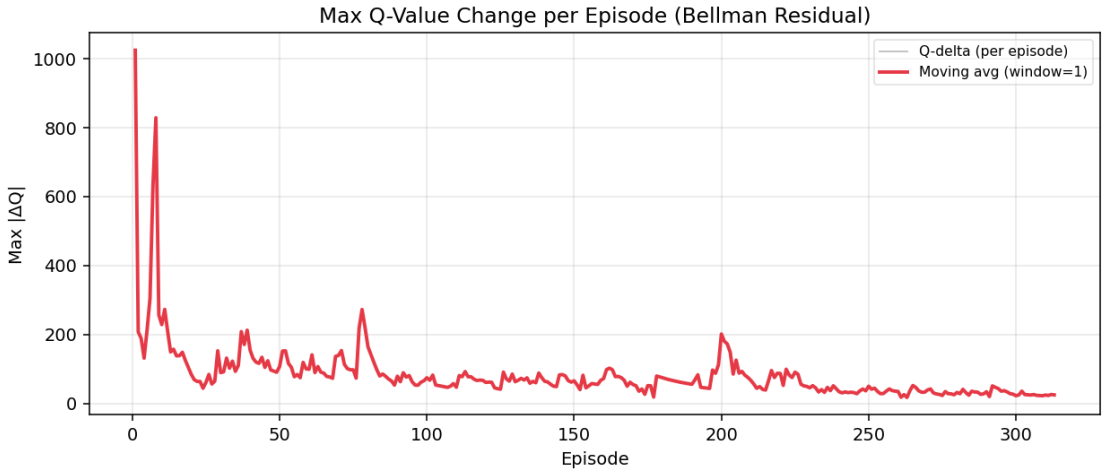


Figure 8: Q-Table change across training episodes in single-agent case.

The ΔQ begins as high as 1024 at episode 2, before converging down to the neighborhood $22 - 25$ as shown in Figure 8. Consistent with Q-delta stabilisation, action entropy declined

from approximately 0.775 to 0.62 per Figure 2, indicating the agent's behaviour concentrated over a narrowing subset of actions as Q-values self-consistently converged.

Comparison: Multi Agent Case

The three agent case demonstrates that $\text{BRGap}_{\text{norm}}$ reached 0.282 during the final evaluation, which is above the 0.25 threshold. This indicates that the joint policy did not satisfy the Nash ε-equilibrium criterion under this metric.

Note that ΔQ convergence was nonetheless achieved, declining from 1600 to 30 9. The divergence between algorithmic convergence and Nash approximation quality can be attributable to two factors. The first factor is the increased complexity of the multi-agent retrain procedure. During this procedure, each agent's best-response is approximated independently against frozen opponents. The second factor is market non-stationarity, which inflates retrain-based BRGap estimates where episodes span multiple market regimes [28]. To address these issues, further design refinement was implemented as per Section 4.7.

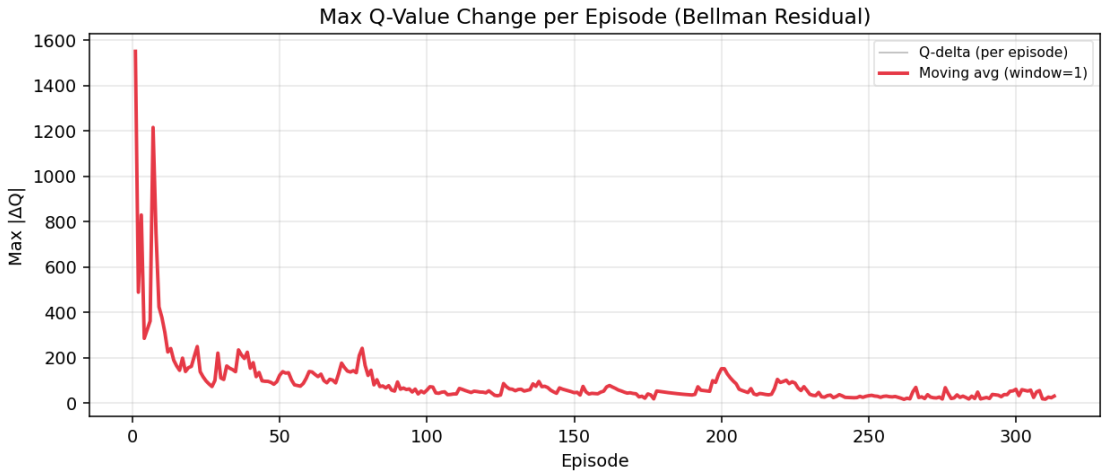


Figure 9: Q-Table change across training episodes in three-agent case.

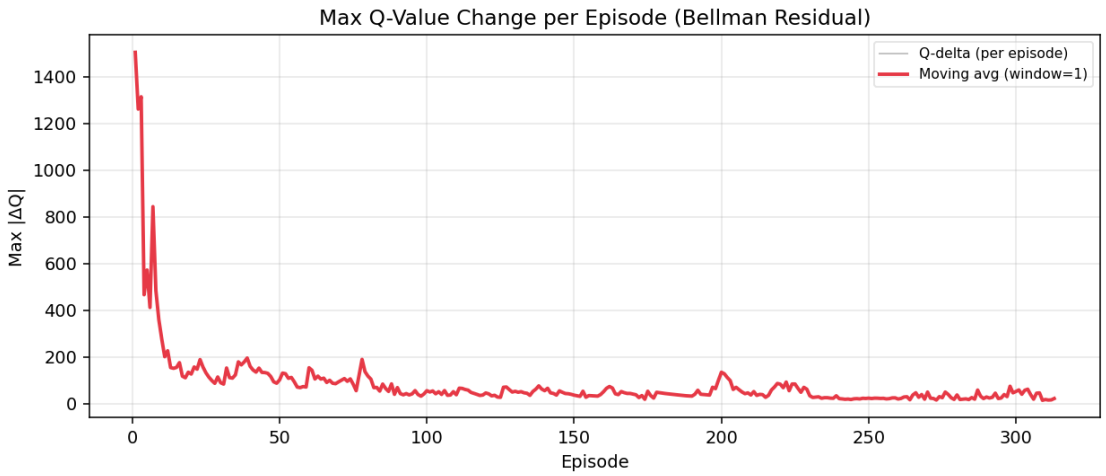


Figure 10: Q-Table change across training episodes in five-agent case.

The tabular Q-learning framework scales independently across agent counts because each agent maintains its own Q-table over an identical state-action space, with no joint-action enumeration required. Adding further RL agents does not expand the per-agent table or alter the update rule. Instead, it introduces additional environmental non-stationarity. The primary scaling cost is in opponent simulation overhead, which grows linearly with agent count. Beyond five agents, the binding constraint is the training data horizon discussed in Section 4.1. The 14-day rolling window and ERCOT data ceiling limit the number of distinct episodes available to each agent in the multi-agent framework. The empirical effect of agent count on cumulative reward is examined in Section 4.5. Note that further discussion on training convergence is discussed in 6.

4.4 Single Agent Baseline Results

The policy evaluated in this section was selected via the late-save mechanism: training monitored normalized BRGap every 20 episodes beginning at episode 230, and preserved the policy snapshot at episode 260, where the normalized BRGap fell below the 0.25 threshold, indicating no agent could gain more than 25% of its current reward through unilateral deviation at that training checkpoint.

ERCOT Baseline Test

The saved single-agent policy was evaluated against three cost-plus markup baselines over 28 episodes drawn from the 2019-2024 ERCOT market data. The baselines utilized the design in Section 3.3 with fixed markups of 25 %, 30 %, or 35 %.

Policy	Mean Profit ($)	Std ($)	Win Rate (%)	Profit Share (%)	Mean Bid ($/MWh)
Cost-Plus $25	54,215	64,315	25.0	16.31	294.55
Cost-Plus $30	50,396	59,660	7.14	15.20	294.55
Cost-Plus $35	48,027	56,544	14.29	14.55	294.55
RL Agent	**70,044**	110,668	**60.71**	**53.94**	**94.99**

Table 3: Single-agent ERCOT baseline comparison over 28 evaluation episodes.

The RL agent achieves a mean episodic profit of $70,044, outperforming the best cost-plus baseline ($25 markup, $54,215) by 29.2%. This improvement is driven primarily by a fundamentally different bidding strategy. The RL agent bids at a mean price of $94.99/MWh at approximately 20.2 MW, compared to the cost-plus agents which bid at $294.55/MWh at 46.9 MW. By submitting lower, more competitive bids, the RL agent achieves a win rate of 60.71% and captures 53.94% of total episode profit, while no single cost-plus agent exceeds a 16.31% profit share.

The trade-off of this strategy is reflected in the reward variance. The RL agent exhibits a standard deviation of $110,668 compared to approximately $60,000 for the cost-plus baselines, and a minimum episodic profit of $-\$7,071$. This higher variance is consistent with the adaptive nature of the learned policy, which concentrates bids in a narrower price-quantity region and is therefore more sensitive to clearing price fluctuations than the diversified cost-plus strategy.

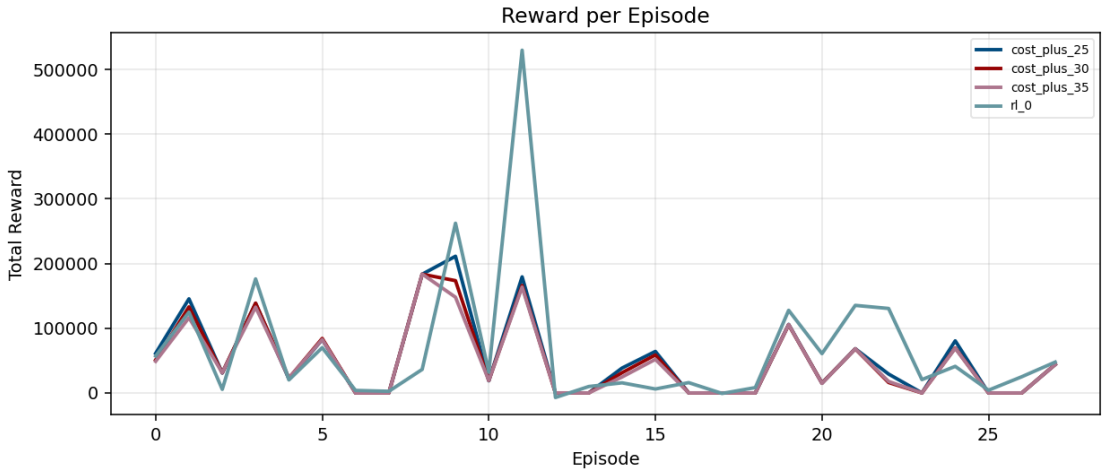


Figure 11: Episode rewards in single-agent case compared to cost-plus markup baselines over the 2019-2024 ERCOT evaluation period.

4.5 Multi-Agent Baseline Results

In the multi-agent setting, the saved policies from the three-agent run and five-agent run were each evaluated against the same three cost-plus markup baselines over 28 episodes. In both cases the late-save threshold was not triggered during training; the final episode policy was used instead.

Three-Agent ERCOT Baseline Test

Policy	Mean Profit ($)	Profit Std ($)	Win Rate (%)	Profit Share (%)	Mean Bid ($/MWh)
Cost-Plus 25 %	51,086	61,127	10.71	10.45	294.55
Cost-Plus 30 %	49,091	58,893	3.57	10.00	294.55
Cost-Plus 35 %	45,872	53,992	3.57	9.35	294.55
RL Agent 0	56,353	61,714	35.71	27.94	64.70
RL Agent 1	**74,392**	92,194	**28.57**	22.60	58.52
RL Agent 2	66,661	86,542	17.86	19.65	63.66

Table 4: Three-agent ERCOT baseline comparison over 28 evaluation episodes.

All three RL agents individually outperform all three cost-plus baselines on mean episodic profit, collectively capturing 70.19% of total episode profit against the cost-plus agents' combined 29.80%. The RL agents bid at substantially lower prices ($58-$65/MWh) than the cost-plus agents ($294.55/MWh), reflecting a learned strategy of competitive underbidding to secure market clearing. The introduction of inter-agent competition among RL agents reduces individual win rates relative to the single-agent case, yet each RL agent's win rate (17.86-35.71%) still substantially exceeds any cost-plus agent (3.57-10.71%).

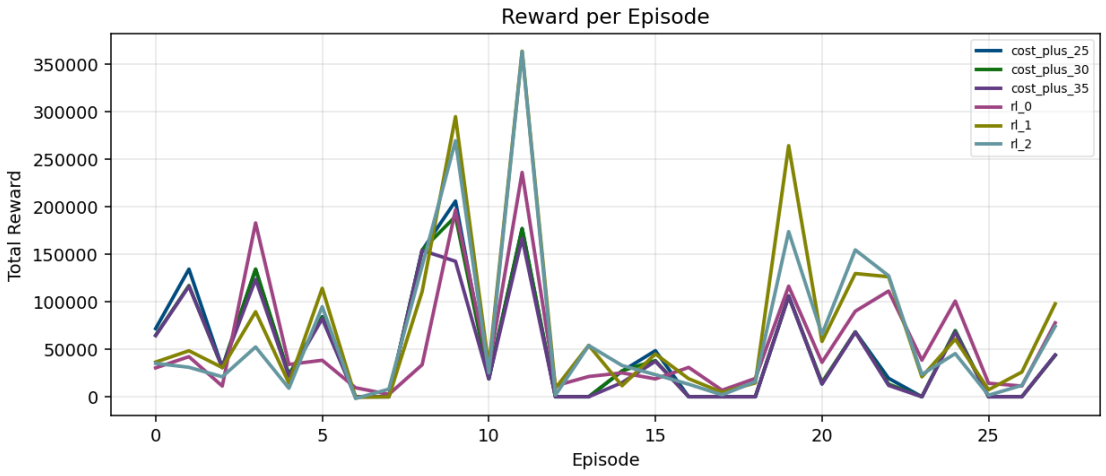


Figure 12: Episode rewards in three-agent case compared to cost-plus markup baselines.

Five-Agent ERCOT Baseline Test

Table 5: Five-agent ERCOT baseline comparison over 28 evaluation episodes.

Policy	Mean Profit ($)	Profit Std ($)	Win Rate (%)	Profit Share (%)	Mean Bid ($/MWh)	Min ($)	$CVaR_{10}$ ($)
Cost-Plus 25 %	44,814	51,638	7.14	7.49	294.55	0.00	0.00
Cost-Plus 30 %	44,488	50,992	0.00	7.48	294.55	0.00	0.00
Cost-Plus 35 %	43,664	49,461	3.57	7.40	294.55	0.00	0.00
RL Agent 0	54,373	68,429	17.86	15.64	63.82	2,034	3,206
RL Agent 1	**80,295**	98,088	**42.86**	20.94	81.68	2,048	2,285
RL Agent 2	56,320	65,311	3.57	13.43	78.75	1,380	1,930
RL Agent 3	63,471	91,224	0.00	13.82	79.90	−2,265	−159
RL Agent 4	71,663	108,648	25.00	13.80	60.64	−806	2,092

In the five-agent competitive setting, all RL agents continue to outperform all cost-plus baselines on mean episodic profit. The best-performing RL agent achieves $80,295, exceeding the best cost-plus baseline by 79.17%. Notably, the cost-plus agents record zero minimum profit and zero CVaR across the evaluation window, reflecting episodes where they failed to clear the market entirely. By contrast, RL agents maintain positive minimum profits in three of five cases, with CVaR values indicating bounded downside risk. Increased competition among five RL agents compresses individual bid prices further relative to the three-agent case, consistent with the competition adjustment mechanism described in Section 3.2.

4.6 Additional design iterations

Outside the parameter selection process described in Section 4.2, our team attempted to address questions surrounding expanding market conditions, and multi-agent convergence. As a result, the team implemented more data, updated parameter selection, and modified threshold methodology.

Dataset Expansion and Episode Horizon

As part of an iterative design approach, the training dataset to the period January 1st, 2018 through December 31st, 2025, yielding 2,555 available episode start positions. This represented an approximately eight-fold increase.

This initial design decision was inspired by a few reasons. A single calendar year captures one seasonal cycle and one set of fuel-price and renewable-penetration conditions, leaving the Q-table without exposure to high-price regimes such as Winter Storm Uri [28]. As discussed in Section 4.3, the multi-agent convergence through BRGap was not observed under the parameter selection. By including a larger training dataset, further observations would help thep policy's converge to a Nash Equilibrium [12].

Parameter Changes: Temperature

With the expanded training horizon, the original exponential decay schedule reaches the temperature floor at approximately episode 461 per Section 4.2, representing only 18% of the 2,555-episode training window.

The schedule was redesigned with decay = 0.9993 and $T_{\min} = 0.05$, delaying the transition to $T_{\min}$ to episode 2,300. This preserves meaningful exploration across the full dataset. Note that the 5-agent run applied one further adjustment relative to the 3-agent configuration, with a slightly slower decay applied.

Parameter Name	**Value Used - Single Agent**	**Value Used - Three Agent**	**Value Used - Five Agent**
temperature decay	0.9993	0.9993	0.9995
temperature minimum	0.05	0.05	0.05

Table 6: Summary of changes to temperature parameterization applied.

Parameter Selection and Threshold Methodological Changes: Reward, ΔQ, and BRGap

Reward normalization: Per-step rewards were normalized by the maximum notional exposure defined in Section 4.2:

$$\tilde{r}_{i,t} = \frac{r_{i,t}}{\text{max\_notional}}, \qquad \text{max\_notional} = q_{\max} \times \hat{P}_{0.95} \tag{6}$$

where $\hat{P}_{0.95}$ is the 95th percentile of the absolute historical LMP distribution over the 2018 - 2025 training window. For the extended dataset, this evaluates to $\hat{P}_{0.95} \approx \$78.47\,\text{MWh}^{-1}$, giving max_notional $\approx$ \$3,924 [4].

Since the training dataset covers multiple systemic changes and pricing regimes, normalization provides a better relative value that prevents destabilizing Q-value estimates across the state space in the case of major volatility [28].

ΔQ recalibration: With normalized rewards, the ΔQ operates on a reduced scale. The normalized ΔQ converges to values in the range $0.1 - 0.3$. The original threshold of 50, which is calibrated on un-normalized rewards, was permanently inactive. The threshold was rescaled to 0.2, preserving the same conceptual convergence gate on the normalized scale. The updated ΔQ trajectories for the multi-agent cases are shown in Figures 30 to 31.

BRGap methodology: Early multi-agent runs using the `retrain` BRGap method produced a BRGap_{norm} of ≈ 1.03, indicating that a freshly trained best-response agent substantially outperformed the current policy. J_{current} was evaluated under the Boltzmann policy at temperature $T = 0.05$, while J_{BR} was evaluated under a greedy retrain. Under the `greedy q` method, J_{BR} is estimated by having agent i select actions greedily with all other agents following their frozen policies. $\text{BRGap}_{\text{norm}}$ is then computed using this greedy J_{BR} via Equation 4. Since the Softmax policy converges to the greedy policy as $T \to T_{\min}$, BRGap $\to 0$ naturally, providing a computationally efficient convergence signal that avoids the cost of retraining a separate best-response clone at every checkpoint. The pseudocode for the `greedy q` method is given in Algorithm 1.

The multi-agent case BRGap trajectories with the updated design are shown in Figure 27

and Figure 28. For both the three and five agent cases, the 0.25 threshold was satisfied at episode 2,501, consistent with the harder convergence problem posed by simultaneous opponent non-stationarity as mentioned in Section 4.3 [12].

In the single-agent case, $\text{BRGap}_{\text{norm}}$ first crosses the 0.10 threshold at episode 2,251, triggering the late-save mechanism. This is shown in Figure 26. Table 11 summarizes the parameters utilized as part of the model on the extended training run.

In-Sample and Sequential Baseline Comparisons

Policies saved at the late-save trigger episodes were evaluated against the same three cost-plus markup baselines used in Sections 4.4 and 4.5. Additionally, the evaluation window spans 2019-01-01 to 2024-12-31, with $k = 28$ randomly drawn episodes also matching the previous training runs. It is important to note that the 2019 - 2024 evaluation period falls within the training distribution for updated runs. Reward per episode for each baseline can be found in Figures 32, 33, and 34.

To isolate the effect of design changes, all six policies were additionally evaluated on the same 14 sequentially drawn episodes from December 2025 using sequential episodes. Sequential sampling ensures each policy faces identical market conditions, enabling direct cross-design comparison within each agent-count tier. December 2025 was held out of sample for both the original design and iterated design, representing a fair testing period. Figure 38 shows the sequential comparison results.

4.7 Sensitivity Analysis

Figure 25 presents cumulative reward under two and five RL agent configurations, complementing the parameter selection ablations in section 4.2 . Per-agent reward remains positive across both agent counts, consistent with the architectural scalability argument made in section 4.3.3. The modest decline in per-agent reward at five agents is expected under increased competition for cleared quantity. The elasticity perturbation results are discussed in section 5.1. The policy freeze interval and its effect on exploration under multi-agent non-stationarity are examined in depth in section 6.2.

5 Model Verification and Validation

After a Q-learning policy has been trained in the simulated market, we must determine whether the resulting policy is reliable and useful. Verification means checking if the implementation behaves consistently with the model being built. Validation means determining if the learned policy remains competitive when compared to other policies.

5.1 Policy Evaluation Against Baselines

The learned policy π_{RL} is compared against baseline policies that have non-learning rules, including fixed-structure and historical-statistics-based approaches. These baselines provide

a reference point because if they cannot beat simple baselines, they are not justifiable, and also they reduce the risk of over-claiming reinforcement learning success.

Performance is estimated empirically by averaging cumulative episode reward across multiple random seeds and evaluation episodes. Using several seeds matters, because relying on a single run can give a distorted impression. This is why reporting the mean and spread across seeds gives a more reliable picture of the policy quality. Policies are evaluated under matched random conditions so that seed-level differences are less significant. If the average advantage is positive and statistically significant, then this is greater evidence of improvement.

Looking at mean return by itself can be misleading. A policy may perform well on average while still producing highly variable or tail-heavy outcomes. This is why it is important to report spread and downside behaviour as well as the mean, for example through reward variance, Sharpe-like ratios, or conditional value at risk applied to episode-level profit. A policy that achieves higher mean reward but with significantly worse downside risk is not necessarily better for a generator operator who faces extreme risk in bad episodes [23, 1].

Robustness under perturbation implies a good policy should still be good when the environment faces minor changes. In particular, one may ask whether the ranking of π_{RL} relative to the baselines is preserved when key model inputs are stressed. If policy rankings change dramatically under mild perturbations, then the performance claims are not as strong because it shows that the policy is overfitted. On the other hand, if the tests show stability, this supports the validity of the model.

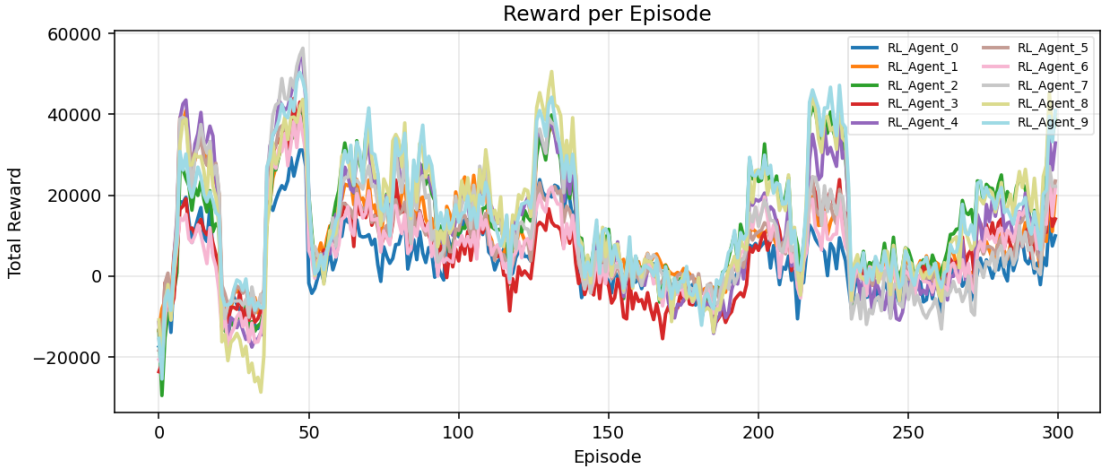


Figure 13: Aggregate RL reward across episodes.

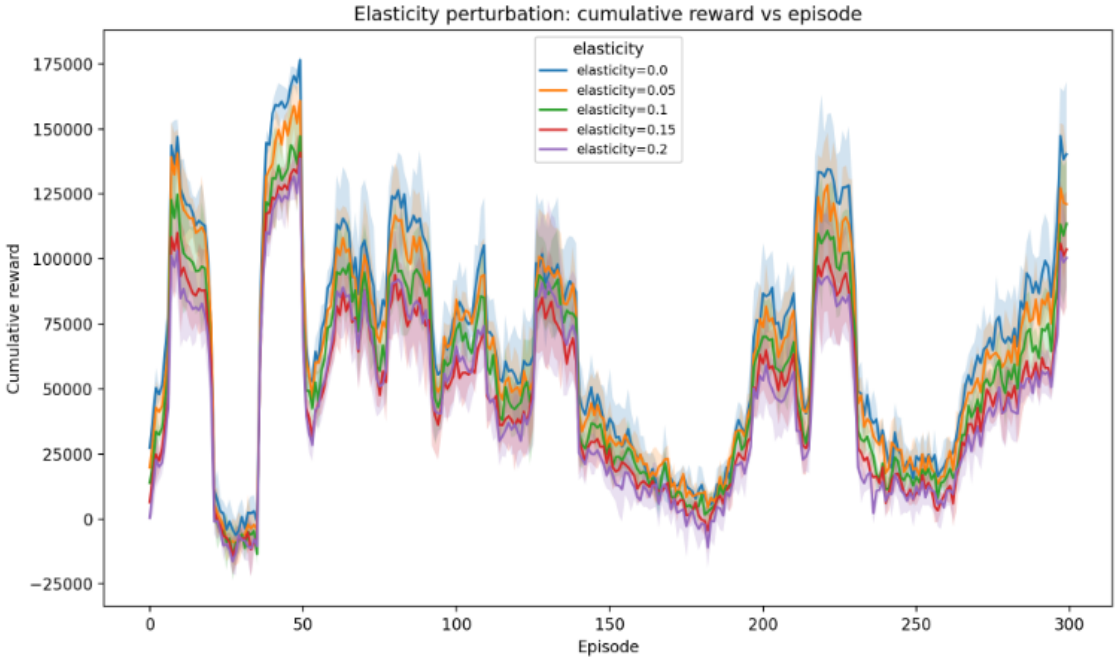


Figure 14: Sensitivity of cumulative reward to the competition-elasticity assumption.

5.2 Search / Verification Results

Even in tabular form, the number of deterministic policies grows exponentially, so an exhaustive search over all policies is infeasible. For this reason, we use search-based verification to determine whether the learned policy is locally improvable in a meaningful way.

A finite comparison library is formed from the learned policy, the two non-learning baselines (cost-plus markup and historical quantile), and a set of structured perturbations of the learned policy. For example, versions produced under different exploration freeze intervals, warm-start configurations, or demand-scale conditions. Warm starting is relevant here because it initializes the Q-table using cost-plus markup returns before training begins, which anchors early learning to a reasonable operating point and means the perturbations around the learned policy are being compared against a policy that already had a sensible starting point. All policies in this library are evaluated on the same episode set under the same random seeds. The best policy found in the library is identified by mean cumulative reward, and the gap between that best policy and RL is recorded. When this gap is small enough, it suggests that RL is close to the best policy in the tested class. This should not be interpreted as a claim of global optimality over all admissible policies. It says that within the reduced library being checked, the learned policy is difficult to improve.

To make a stronger claim about convergence toward a good policy, the framework draws on the concept of weak acyclicity from Arslan and Yüksel [2]. A stochastic game is weakly acyclic if, from any starting joint policy, there exists a sequence of best-reply updates that leads to an equilibrium. Verifying that the learned policies follow convergence paths consistent with weak acyclicity supports the claim that the multi-agent training process is not just

stabilizing arbitrarily, but is moving toward something strategically meaningful.

The natural endpoint of that convergence is a Nash equilibrium, where no individual agent can improve its own reward by unilaterally changing its policy while the others hold theirs fixed. In practice this is assessed through the best-response gap, which measures how much profit an agent could gain by switching away from the learned policy against fixed opponents. When the best-response gap is small across agents, the learned joint policy is close to equilibrium, and the verification gap described above and the best-response gap are therefore measuring compatible things from different angles.

Late-stage policy behaviour introduces an additional source of uncertainty. Policies evaluated near the end of training can differ depending on which random seed was used, since the final episodes are a relatively small sample and the Q-table may have settled into slightly different regions across runs. For this reason, late-stage policies are compared against randomized-seed baselines to distinguish genuine convergence from seed-specific outcomes. Value-function consistency is assessed by examining whether the Q-values recorded at the end of training assign higher values to state-action pairs that produce higher rewards. In a well-converged tabular model, Q-values should approximately equal the one-step reward plus a discounted continuation value. If states that consistently produce high reward are assigned low Q-values, this would show there is a problem with the training procedure, not just a suboptimal policy.

Taken together, these checks support the following modest verification claim: the learned policy outperforms meaningful baselines on average return, gains persist under perturbation and risk-aware views, local policy search finds limited improvement opportunities, and learned values are directionally consistent with observed rewards in relevant operating regions. Therefore, no claim of universal optimality is made here. The implementation should behave consistently under the checks used in this section and the resulting policy should be practically competitive in the modeled bidding environment.

Taken together, these checks support the following modest verification claim:

1. The learned policy outperforms meaningful baselines on average return.
2. Gains persist under perturbation and risk-aware views.
3. Local policy search finds limited improvement opportunities.
4. Learned values are directionally consistent with observed rewards in relevant operating regions.

Therefore, no claim of universal optimality is made here. The implementation should behave consistently under the checks used in this section and the resulting policy should be practically competitive in the modeled bidding environment.

6 Convergence and Learning Behaviour

This section shifts focus from whether the learned policy is competitive to how it got there. Training a Q-learning agent is not just about running enough episodes, it matters whether the learning process is actually settling toward something stable and meaningful rather than just wandering.

6.1 Q-Value Stabilization

This section focuses on how learning moves from exploratory fluctuation toward more stable policy behaviour. In tabular Q-learning, as seen in Section 3.1, each update takes the form:

$$Q_{i,t+1}(s_t,\, a_{i,t}) = Q_{i,t}(s_t,\, a_{i,t}) + \alpha_{i,t}\Big[r_{i,t} + \gamma \max_{a'} Q_{i,t}(s_{t+1}, a') - Q_{i,t}(s_t,\, a_{i,t})\Big],$$

As state-action visitation increases and exploration decays, the sequence $\{Q_{i,t}\}$ should show the reduced variability in frequently visited regions.

From stochastic approximation theory [2], convergence in finite MDPs is supported when the step sizes satisfy:

$$\sum_n \alpha_n = \infty, \qquad \sum_n \alpha_n^2 < \infty,$$

where n indexes successive visits to a given state-action pair. In this project, the learning rate is updated harmonically as $\alpha_{i,t}(s,a) = \alpha_{i,0}/N_{i,t}(s,a)^{\xi}$, where $N_{i,t}(s,a)$ is the visit count for pair (s,a) up to time t and $\xi \in (0.5, 1]$ is the learning rate exponent described previously. This schedule satisfies both conditions: the sum of $1/n$ diverges while the sum of $1/n^2$ converges. Exact convergence conditions are only approximately met in finite-horizon simulation. Consequently, convergence is evaluated through stabilization diagnostics, not strict theorem conditions.

We track three indicators over training. The first is update magnitude: the size of each incremental Q-table correction, namely the absolute temporal-difference update, should shrink over time as the agent accumulates experience and the learning rate decays.[22] A failure to shrink would indicate that the reward signal or the exploration schedule is preventing the table from settling.[26]

The second is greedy policy switching rate: the fraction of states for which the greedy action changes between successive episodes. This is used here as a convergence diagnostic rather than as a standard benchmark metric. It should decrease as the policy stabilises; high late-stage switching would suggest either that the table has not converged or that the multi-agent market environment is too non-stationary for the current agent design to track reliably.[2]

The third is temporal-difference consistency: for a well-converged policy, the stored Q-value for a visited state-action pair should approximately satisfy the Bellman consistency relation, meaning it should align with the observed one-step reward plus the discounted continuation value at the next state.[31, 22] Systematic deviations from this relationship indicate that the

agent's value estimates have not yet aligned with the observed market dynamics.

Because each indicator can be misleading when viewed on its own, stabilization is only convincing when all three show sustained improvement. For example, a shrinking update magnitude alone could indicate premature loss of exploration, while a stable greedy policy could still exist in a poorly performing region of the action space.

Figure 15 shows the harmonic decay of a single agent's effective learning rate, dropping by roughly three orders of magnitude over training. Figure 16 complements this with the saved entropy and rolling-variance traces, together with the mean-reward series taken from the same logged horizon.

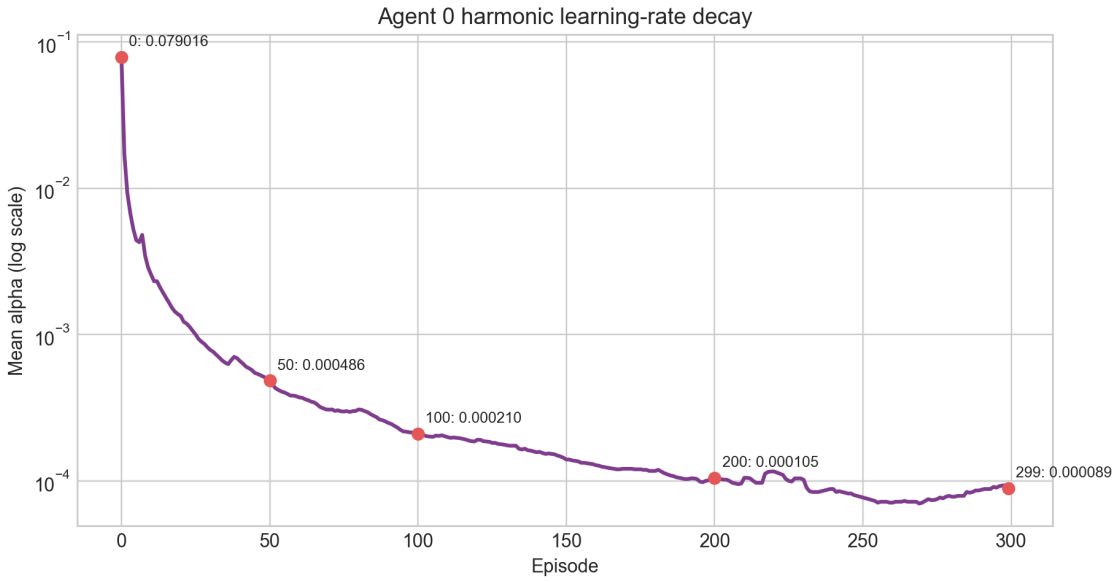


Figure 15: Harmonic decay of a single agent's mean learning rate across training episodes on a log scale. This is consistent with progressively smaller Q-value updates late in training.

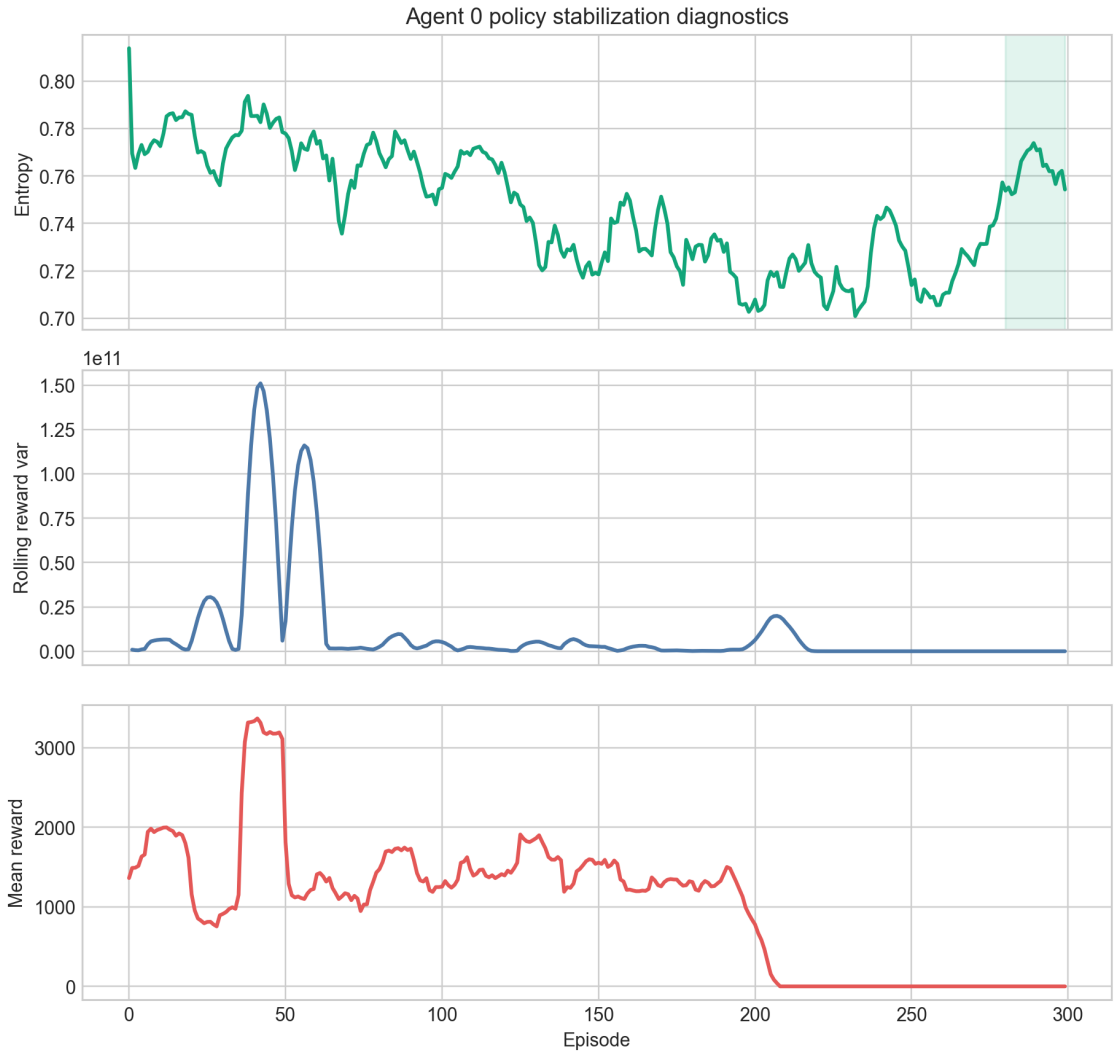


Figure 16: Available stabilization diagnostics for a single agent: action entropy, rolling reward variance, and mean episode reward.

The broad pattern is consistent with a training process that is settling down. Early episodes show larger effective updates and more exploration. In mid-training, the most frequently visited states begin to exhibit more stable action preferences, while sparse states remain noisy. Late training is characterised by a much smaller learning rate and flatter saved stability traces, which is the type of practical convergence behaviour one would expect from a decision-support model of this kind.

Finally, stabilization should not be read in isolation from outcome quality. A policy may stabilize to a poor local region if exploration or reward shaping is inadequate. For this reason, the convergence discussion here is paired with the baseline and perturbation results rather than treated as evidence on its own.

6.2 Policy Freezing and Exploration Effects

A research paper that the group read on Q-learning in multi-agent settings introduced the idea of policy freezing and explained why it is useful when multiple agents are learning at the same time[2]. The core issue in multi-agent Q-learning is that since all agents are updating their policies simultaneously, the environment looks non-stationary from any single agent's perspective, meaning the world each agent is trying to learn about keeps changing because the other agents keep changing too. This can prevent Q-learning from converging to a stable solution altogether. Policy freezing addresses this by periodically locking in each agent's policy for a set number of episodes, making the environment appear more stationary from each agent's point of view and giving Q-values a chance to stabilize. Every so often, the agent pulls a greedy policy out of its current Q-table and locks it in for a set number of episodes. During that time, Q-values keep getting updated in the background, but the agent's actual behavior stays the same. Once the window is up, the policy gets refreshed using the latest Q-values.

The training loop maintains, for each agent, a deterministic policy mapping from discretized state keys to action indices. At the beginning of each episode whose index is a multiple of the user-specified freeze horizon K, the algorithm extracts a near-greedy policy from the Q-table using a very small softmax temperature and merges it with the previous frozen policy. A policy inertia parameter controls how likely the agent is to retain its previous action for a given state versus adopting the newly suggested greedy action. Between policy updates, the agents behave according to these frozen policies while Q-values continue to be updated based on realized rewards. When policy freezing is disabled or $K = 1$, the behaviour reduces to standard online Q-learning with per-step exploration governed by the episode temperature.

6.3 Application to the Electricity Market Setting

The developed framework is customized for ISO's, where bidding takes place in the day-ahead market, and an ample amount of underlying data infrastructure exists. This framework allows us to conduct multi-agent reinforcement learning using a realistic market context, where demand, prices, and the opposition stack are based on real historical market data as opposed to toy market data. Historical Henry Hub natural gas price data is used to construct the marginal cost distribution in \$/MWh, where each agent gets a realization of marginal cost at each iteration. The reward is then calculated as the quantity cleared by the agent, multiplied by the difference between the price that the fuel cleared at and marginal cost for that agent, minus a penalty for bids that exceed exposure limits. Load forecasts combined with historical LMP behavior, fuel mix data, generator inventories, market-based marginal costs, and opposition stacks give us an approach that retains all the important qualitative aspects of reality without being intractable to tabular Q-learning.

7 Implications of Results

7.1 Market Power and Generator Strategy

Table 7 captures the baseline results as discussed in Section 4.4 and Section 4.5, across the 1, 3, and 5 agent cases.

Table 7: Mean per-agent cumulative reward ($) for the RL policy and cost-plus markup baselines, evaluated over the 2019-2024 ERCOT test period.

Configuration	RL (mean)	CP+25%	CP+30%	CP+35%	RL (mean)/ CP+25%
1-agent	$70,044	$54,215	$50,396	$48,027	+29.2%
3-agent (median agent)	$66,661	$51,086	$49,090	$45,872	+30.49%
5-agent (median agent)	$63,471	$44,814	$44,487	$43,664	+41.63%

The 1-agent result, with no RL competition, is a direct comparison with the three cost-plus markup strategies. Extrapolating the results in Section 4.4 over a full year results in the RL policy capturing $\approx \$420,000$ more in revenue per Table 7. Noting the discussion in 4.4, since the agent bids at a lower, variable price, it is more competitive compared to the baselines with a 60.71% win rate and 53.94% profit share. As shown in Figure 17, the agent learns to track the merit order rather than applying a constant percentage lift, bidding at or above near marginal cost when it is the clearing unit and holding back when it is not the clearing unit. These two types of bidding curves are commonly deployed by power generators in actual ISO markets [21].

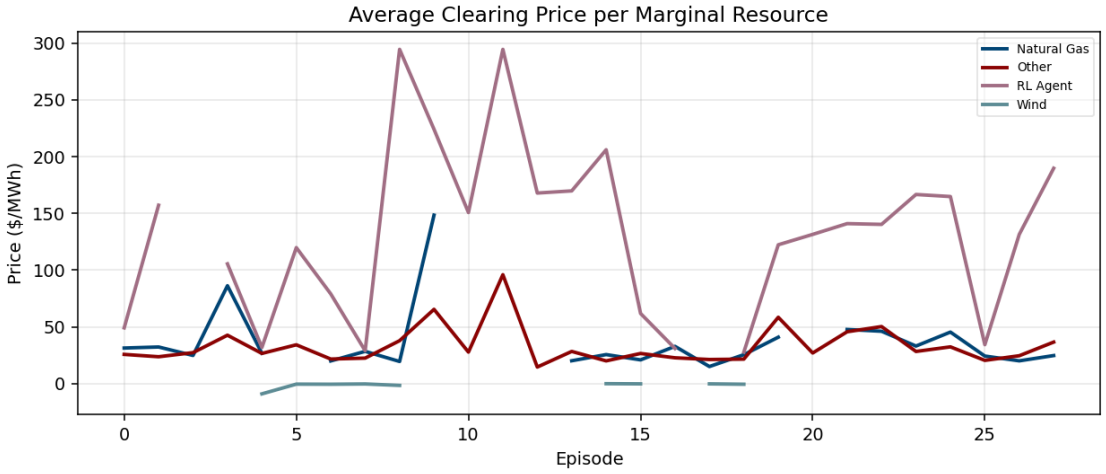


Figure 17: Clearing price for each Marginal Resource across each evaluation episode in single agent case.

The multi-agent results show that per-agent advantage persists under RL competition, with Table 7 demonstrating a 30.49% and 41.63% improvement from the median RL agent's in the 3-agent and 5-agent environments over the 25% cost plus markup agents. Additionally, Figure 11 demonstrates consistent out performance within the 3-agent. It is important that

the baseline agents do not adapt, with the 25% cost plus markup agent in the the 1-agent, 3-agent and 5-agent environments producing identical mean bidding price and quantity.

The sequential evaluation on December 2025 discussed in Section 4.6 allows a direct cross-design comparison. Table 19 shows that the extended 2018–2025 design substantially narrows the RL/cost-plus profit gap in multi-agent settings. The ratio of mean RL profit to CP+25% profit improves from 57% to 74% in the 3-agent case and from 50% to 68% in the 5-agent case, while the 1-agent case shows marginal change. The extended training run does not produce a dominant agent, as the cost-plus agents retain the profit lead in December 2025 across all configurations. The source of improvement is equilibrium stability rather than individual dominance. Agents trained on broader market conditions learn lower, more competitive bids, which is corroborated by mean RL bid falling from \$119–\$192/MWh to \$56–\$84/MWh across the multi-agent cases. This reflects convergence to near-greedy strategies validated by the BRGap criterion given in Section 4.6.

In North American ISOs, participants range from sophisticated algorithmic bidders to smaller generators operating on fixed cost-plus rules [18] [29]. The profitability advantage demonstrated within the simulation is plausibly achievable in practice by any market participant that adopts an adaptive strategy against a less sophisticated field.

7.2 Market Mechanism Design

Table 12 summarizes the RL bid and clearing price across the 1, 3, and 5 agent configurations. Table 18 extends this analysis to the updated 2018-2025 design evaluated on the same in-sample period. Table 20 reports both designs evaluated on the sequential evaluation framework. It is important to note that Mean RL Bid Efficiency is defined as the fraction of dispatch intervals in which the submitted bid price exceeds the realized market clearing price, and the agent is not dispatched. Additionally, Δ_{clear} is the change in clearing price relative to the 1-agent baseline.

Across the original design and iterated design, mean bid prices and clearing prices decline as agent count increases. In the original design 12, the average RL bid exhibits a parabolic shape, falling from \$94.99 at 1 agent to \$62.29 at 3 agents before recovering to \$72.96 at 5 agents. In the updated design 18, the bids show a more drastic parabolic shape, going from \$73.73 to \$60.77 to \$72.33 between the cases.

The more consequential difference lies in bid efficiency. The original design bids more aggressively, with 67 % - 71 % of bids above the clearing price of intervals, while the updated design only does so in 39 % - 45 % of intervals. This shift reflects a volume-over-margin strategy in the updated design, where price-aware state encoding allows agents to accept dispatch more frequently rather than holding back for selective clearing opportunities.

The clearing price response to additional RL agents also differs substantially across designs. In the original design, clearing prices fell by -12.9% and -15.1% on the 3 and 5 agent cases 12, with the sequential evaluation yielding at -18.78% and -20.57% reductions 20. In the

updated design, clearing prices fell by only -3.85% and -4.64% in Table 18, and -7.52% and -8.90% in the sequential evaluation 20. The original design's larger clearing price depression follows from its higher bid efficiency. Agents holding back at elevated prices more frequently occupy the price-setting position at the top of the merit order. The updated design's agents, accepting dispatch more broadly, occupy lower merit-order positions and exert less influence on the clearing margin. In both cases, adding more participants produces a downward clearing price trend, consistent with increased competitive pressure on the merit order [18].

The addition of RL agents in the simulation is structurally analogous to electricity market deregulation, where incumbents are joined by new adaptive participants. An example is the deregulation of the Ontario electricity market in May 2002 [27]. Power prices initially fell as new entrants competed. However, with one entrant retaining control of $\approx 70\%$ of available capacity, the market exhibited oligopolistic price behavior, with dominant participants sustaining bids well above marginal cost through strategic capacity and information withholding. This resulted in prices rebounding, and partial re-regulation of the Ontario market by 2004.

Note the previous example has a dominant player by way of capacity. As shown by [24] for the England and Wales pool in the late 1990s, as entrant count grew and average clearing prices fell, two incumbents maintained disproportionate bid markups through bid scheduling rather than capacity withholding. This is the most direct analogue to our simulation results, as our agents hold no capacity advantage, yet Table 5 demonstrates RL agents sustaining bids much higher than the bids of other power asset types with relatively similar clearing prices and marginal costs, as shown in Figures 35 and 36 for the original design, and Figure 37 for the iterated design.

Both real-market cases arrive at the same result. Dominant participants extract surplus in market deregulation through some advantageous strategy. However, the aggregate welfare, or clearing price, changes drastically whether by way of capacity withholding or by strategic bidding scheduling. Policy wise, this indicates that market reform through participant count alone is insufficient, as the Ontario example demonstrates how simply adding participants did not prevent capacity withholding and higher clearing prices. In the England and Wales power pool example, regulatory licence stipulations and mandatory divestiture led to lower clearing prices while sustaining oligopolistic outcomes [24].

Opening markets to more participants leads to an outcome that concentrates surplus among dominant bidders, with dominance arising through capacity concentration [27] or bidding sophistication [24]. Policy interventions that focus solely on participant count, without constraining capacity withholding, risk replicating the Ontario outcome. Our simulation results suggest that the bidding sophistication channel is the more structurally interesting case. With no capacity advantage, adaptive participants converge to dominant strategies despite clearing prices reducing. This implies that market designers who successfully eliminate capacity withholding may still observe oligopolistic surplus concentration among sophisticated bidders. Accounting for this in market design remains an open policy question [17].

7.3 Engineering Ethics Discussion

The simulation demonstrates that an adaptive RL agent captures 53.94% of total episode profit in the 1-agent case against the non-adaptive cost plus markup baselines 4.4. In a real market, deploying such a strategy against less sophisticated participants raises questions about whether information and algorithmic asymmetry constitutes an unfair market advantage. Noting that sophisticated RL-based bidding strategies are labor-intensive to develop, test, and deploy, if large generators or analytics firms have greater access over such tools, the result is a structural information asymmetry that concentrates market power among well-resourced participants. This is consistent with the observations in Section 7.2 regarding dominant participant outcomes. This raises a broader equity question about who benefits from algorithmic market participation.

As discussed in section 7.2, the same bidding strategy that improves generator profitability also depresses clearing prices at scale 12. The ethical framing depends on which effect dominates, as regulators may view clearing price compression as welfare-improving, while also flagging disproportionate profit concentration as a market-power concern. Furthermore, in the 3 and 5 agent cases, no single agent intends to depress clearing prices. Instead, it is an emergent outcome of simultaneous adaptive learning. Engineers deploying multi-agent RL systems in real markets bear responsibility for anticipating these systemic effects, even when individual agent behavior is locally rational and rule-compliant.

Tabular Q-learning produces an inspectable Q-table, meaning every state-action value can be directly examined. In practice, market regulators require generators to justify submitted bids [27]. A fully opaque deep RL policy such as a DQN would not satisfy this auditability requirement, while a tabular Q-Learning approach is more compatible with disclosure obligations.

7.4 Limitations of the Model

The following list describes limitations as a function of design choices throughout the project.

- **Discretized State/Action Space**: As described in Section 2 and 3, our model utilizes a discretized state and action space. Each component of the state and action space is split into 8 buckets. This granularity was chosen to satisfy the computational requirements of the tabular Q-Learning algorithm. However, finer price and bid increments are not included within the model, which reduces our bidding options.
- **Feasibility restrictions on actions**: Agent bids were not only capped by maximum quantity, but also into a feasible action space set. As discussed in Section 4.2, a maximum notional q parameter was set to 0.95. Furthermore, as discussed in Section 3, negative bids were clipped to 0. The logic was to prevent degenerate bids, enforcing "safe" behavior. In physical electricity markets, sophisticated bidding strategies may go outside conventional risk parameters or involve negative bidding.
- **Uniform-price hourly clearing**: As described in Section 3.2, the ERCOT clearing mechanism primarily involves the economic dispatch optimization. Under the Q-

Learning framework, this was a simpler market simulation with faster training and testing times. However, ERCOT day-ahead optimization also has a security-constraint for more granular load and transmission needs [5]. Our clearing procedure loses the security constraint component of the clearing procedure, which would apply a stricter feasible action set.

- **Opponent bids sampled from Gaussian distribution**: Each opponent's bid price was sampled from a Gaussian distribution centered within their respective fuel cost band as described in Section 3. This parameterization was chosen as it is a scalable and interchangeable method to stochastically generate heterogeneous bids. However, this results in the opponent bid distribution being assumed rather than inferred from real market data. This results in the RL agent learning to respond against a synthetic competitor rather than a historically calibrated one for each opponent.

- **RL Agent Marginal Cost modelling via Historical Gas Price Distribution and static scaling factors**: As described in Section 4.1, the marginal cost of each natural gas unit was modelled using the Henry Hub spot price distribution as a proxy for natural gas fuel cost. Plant-specific costs such as heat rate, variable operations and maintenance expenses, and plant startup costs scaled statically. This approach was chosen as Henry Hub prices are a publicly available dataset that encompass the dominant marginal fuel in ERCOT's generation mix [4], providing a data-grounded cost signal without requiring plant-level engineering data. However, static scaling factors assume the relationship between fuel cost and total marginal cost is uniform across all agents. In practice, there is significant heterogeneity on efficiency, and operations and maintenance [29]. This means that our model does not account for cost heterogeneity across opponents, which may overstate RL agent profitability relative to a granular cost stack.

7.5 Looking Ahead in the Industry

As wind and solar penetration grows across electricity markets, and thus the share of non-dispatchable generation increases, real-time demand-supply imbalances are amplified, driving price volatility [28]. Dispatchable resources like natural gas are increasingly called upon to fill the gap between variable renewable output and realized demand, placing greater strategic importance on their bidding behaviour [18]. This makes adaptive bidding by dispatchable resources both more valuable to asset owners and more consequential for market outcomes, as the clearing price is set precisely by the marginal dispatchable unit at the top of the merit order in high-demand, low-renewable periods.

Noting the examples in Section 7.2, it is suggested that bidding sophistication will attract regulatory scrutiny as adaptive tools become widespread [24] [27]. As regulatory responses to adaptive bidding have precedent, market-power screens and bid transparency requirements are likely to be applied as reinforcement learning tools proliferate in day-ahead power asset bidding.

Finally, the transition to function approximation as a natural next step is discussed in Section 8.

7.6 Triple Bottom Line

The three-part triple bottom line framework provides a structured lens for evaluating the broader consequences of deploying adaptive Q-learning bidding strategies in organized electricity markets.

Economic: The central economic result is a consistent and material improvement in generator profitability relative to cost-plus benchmarks. This is provided by the results in Section 4.4 and 4.5, with incremental revenue captured in Table 7. These gains stem from a learned merit-order tracking strategy, consistent with the strategic behaviors documented in empirical studies of real ISO markets [21, 11]. The extended 2018 - 2025 design yields lower per-agent profits in absolute terms relative to cost-plus baselines on the December 2025 sequential evaluation, but improves equilibrium stability.

Social: The social implications of multi-agent RL bidding are two-sided. On the demand side, the entry of RL participants depresses market clearing prices across both designs and all agent configurations. In the original design sequential evaluation, clearing prices fell as much as -20.57% relative to the single-agent baseline. This is consistent with the merit-order effect, whereby competitive entry lowers the price paid by consumers, with downstream benefits for households and small businesses [3]. On the supply side, however, RL participants capture disproportionate profit shares, replicating the dominant-participant dynamics in Section 7.2. As argued in Section 7.3, if adaptive bidding tools are available primarily to well-resourced generators and analytics firms, this structural asymmetry compounds over time, concentrating market power in ways that regulatory frameworks may not fully anticipate.

Environmental: As wind and solar penetration across North American Electricity markets grows, dispatchable resources such as natural gas are increasingly required to balance variable renewable output against realized demand [28, 13]. The market clearing price is set at the margin precisely by these dispatchable units during high-demand, low-renewable periods. An adaptive bidding framework that positions dispatchable generators more competitively, bidding near marginal cost when dispatch probability is high, supports more efficient real-time integration of renewables by ensuring that the merit order reflects true generation costs rather than fixed markups. More efficient dispatch reduces the need for out-of-merit unit commitment and curtailment of renewable resources. Additionally, the clearing price compression documented in Section 7.2 reduces the economic incentive to withhold dispatchable capacity strategically, which can otherwise create artificial scarcity during renewable generation shortfalls and force higher-emissions peaking units into dispatch.

7.7 Standards, Codes of Practice, and Regulatory Compliance

The design of an RL-based bidding framework for organized electricity markets is constrained by a hierarchy of market rules, federal regulations, and reliability standards. The following

evaluates the principal codes applicable to this framework and documents how specific design choices were shaped by compliance requirements.

ERCOT Market Participation Rules: ERCOT's Nodal Protocols govern all offer submissions in the day-ahead market, specifying admissible price ranges, quantity constraints bounded by each resource's registered capacity, and offer format requirements [5]. The action space design in Section 2.3 directly reflects these constraints, as bid prices discretized from historical LMP data ensure a realistic and protocol-compliant range, and quantity bins are capped at a level reflective of average ERCOT generator capacity. The feasibility projection mechanism in Section 4.2 provides an additional enforcement layer, ensuring that no decoded action submitted to the simulated market violates the maximum notional exposure limit.

FERC Anti-Manipulation and Market Behaviour Rules: FERC's anti-manipulation rule prohibits fraudulent or manipulative conduct affecting jurisdictional markets [8]. The multi-agent results in Section 7.2 demonstrate that clearing price depression is an emergent, decentralized outcome of simultaneous Q-learning. Given our design in Section 3, no single agent coordinates with others or withholds information strategically. This is consistent with competitive market conduct under FERC's framework. Furthermore, FERC market behaviour rules require generators to justify submitted bids in response to market-power screens [8]. This auditability property was a deliberate design criterion discussed in Sections 3.4 and 7.3.

7.8 Information Sources and Lifelong Learning

Responsible engineering practice requires critically evaluating procured information for authority, currency, and objectivity before incorporating it into design decisions. This section documents how sources were assessed throughout the project and how that assessment drove methodological revisions.

Authority: Market data were sourced from ERCOT's public data access portal [4] and Enverus' system-wide LMP series, both of which carry institutional authority. The Independent Market Monitor's annual report [18] provides an authoritative external benchmark against which simulation outcomes were contextualized.

Academic references were evaluated for peer-review status and citation impact. The convergence theory underlying the Q-learning framework draws on Arslan and Yüksel [2], published in high-impact venues with extensive replication in the RL literature. The BRGap criterion is grounded in Nash Q-learning framework [12], and the market-power case studies are drawn from peer-reviewed economic journals [27, 24].

Currency: Training data spans January 2018 through December 2025. This range was chosen because a single calendar year provides insufficient coverage of market regime variation: it excludes the high-price events present in the 2018 - 2025 fuel mix series [28]. The dataset expansion described in Section 4.6 was a direct design response to this currency limitation,

identified after reviewing ERCOT market-regime diversity [18]. ERCOT Nodal Protocols [5] were reviewed against the current protocol version to confirm that the feasibility constraints and clearing mechanism in Sections 3.2 and 4.2 remain consistent with live market rules.

Objectivity and Design Revision: Critical re-assessment of sources drove a methodological change. The original BRGap implementation in Section 4.2 followed the definition in [12]. Per the discussion in Section 4.6, the measured gap systematically overstated deviation incentives through an differing comparison. This bias motivated the switch to the `greedy q` method in Section 4.6.

8 Future Work

The current framework provides a strong foundation for RL-based bidding in a simulated day-ahead electricity market, with clear opportunities for further development to improve realism and practical applicability.

8.1 Improved Opponent Modeling

One clear limitation of the current setup is the simplicity of competitor behaviour. In contrast to this, our model's competitors are heterogeneous and strategic: they differ in marginal costs, forecasting skill, risk preferences, and adaptation rates. If opponents are modeled too rigidly, a learning agent may overfit to that structure and lose performance when conditions change.

A practical intermediate step is opponent-class conditioning: train policies against a diverse mixture of opponent types and evaluate worst-case and average-case outcomes separately. The goal is to ensure that the reported performance advantage of the RL policy does not disappear when the composition of opponents changes. Even without full equilibrium analysis, this improves reliability of performance claims.

8.2 Expanded State and Action Spaces

The current discretization makes tabular learning manageable and keeps the policy interpretable, but it also compresses information. A richer state description incorporating additional load-shape information, fuel-price interactions, or congestion proxies could improve bid quality.

Coarser state aggregation reduces variance and compute cost, but increases bias by merging strategically distinct market conditions into the same bucket. Finer aggregation reduces approximation bias but increases sample complexity and sparsity, making reliable Q-value estimates harder to obtain within a finite training horizon. Future work should make this trade-off explicit through systematic grid-refinement studies and out-of-sample performance tracking.

Action-space refinement is also important. Realistic bidding often requires more flexibility than a small finite set of price-quantity choices. Extending toward structured multi-block bids can increase expressiveness, though it also expands the optimization burden during learning. Hybrid approaches may be a good compromise.

8.3 Function Approximation / Deep Reinforcement Learning

As the state and action representations become richer, tabular methods become increasingly impractical, motivating the use of function approximation to generalize across large state-action spaces [22]. Function approximation offers a way to generalize across unseen states and to reduce reliance on manual discretization. In a deep Q-network approach, a neural network with trainable parameters is fitted by minimizing a squared Bellman-error-style objective over minibatches drawn from a replay buffer of observed transitions, while a periodically updated target network is used to stabilize training [16]. This removes the requirement for explicit binning of continuous state variables and allows the agent to share statistical strength across similar ISOs.

Additional performance comes at the cost of interpretability and stability. A tabular Q-table allows every state-action value to be inspected directly, which is not possible with a neural network approximator. If future work extends the current framework to Deep Q-Networks, that transition should be supported by strict ablation studies, uncertainty and sensitivity analysis across seeds and hyperparameters, and diagnostics for policy explainability in economically meaningful terms. Another promising extension is risk-aware DQN, where the objective explicitly penalizes downside exposure, improving operational relevance in volatile pricing regimes.

8.4 Additional Market Features and Constraints

To move from a prototype simulation towards something trustworthy enough to use in practice, the environment should incorporate richer operational and market constraints. Some examples of this include tighter ramping behaviour, settlement or imbalance penalties, and additional reliability-oriented operating rules. These additions would also bring the simulator closer to the ERCOT real-time and day-ahead structures that the historical data ultimately reflects.

It is important to note that useful performance does not require every part of the table to converge equally well. Electricity-market operation is concentrated in a smaller set of recurrent states, and stable decisions in those regions matter more than highly polished estimates in rarely visited edge cases. For this reason, visitation-weighted stability summaries would be a useful extension to the current diagnostics.

This also has an ethics and governance dimension. High-performing bidding strategies should be assessed not only by expected return but also by fairness, market-power implications, and regulatory compatibility. Embedding these considerations directly in model constraints and evaluation metrics would strengthen both technical and professional credibility.

9 Conclusion

The thesis proposes a tabular Q-learning model for generator bidding decisions in the day-ahead electricity market. The bidding problem was formulated as a Markov Decision Problem (MDP). The state space is defined by the history of demand, locational marginal prices, a 14-day demand prediction, and the time-of-day indicator, while the action space is discrete and consists of bid-price/quantity combinations. Tabular Q-learning was chosen because it offers greater interpretability than other techniques such as CAPM-based optimization or simulation-based optimization.

9.1 Results

The learned policy was evaluated against cost-plus markup baselines of 25 %, 30 %, and 35 % over 28 episodes drawn from the 2019 - 2024 ERCOT evaluation period. As shown in Section 4.4 and Section 7.1, in the single-agent case, the RL agent achieves better performance over the baselines. This advantage was driven by a fundamentally different bidding strategy. Instead of applying a fixed markup, the agent learned to bid based on expected clearing conditions. It bid more selectively and priced close to marginal cost when there was a higher probability of clearing and withholding capacity otherwise.

This performance advantage persisted under competition. When we had three agents, the median RL agent outperformed the best cost-plus baseline by 30.49%. All three RL agents individually outperformed all three baselines and captured 70.19% of total episode profit collectively. When we had five agents, the median RL agent outperformed the best cost-plus baseline by 41.63%, and the best-performing RL agent achieved \$80,295 against the best cost-plus baseline of \$44,814. However, as shown in Section 4.5, some of the agents in the five-agent setting failed to clear the market in some episodes resulting in zero profit.

The size of the Q-value updates got smaller over time with training. Updates were initially large at 1,024 in the single-agent case and 1,600 in the three-agent case and stabilized to neighbourhoods reported in Tables 1 and 8. In the single-agent design, the BRGap threshold was reached by episode 260, while in the multi-agent design, BRGap was not met. This motivated design improvments, detailed in Section 4.6, including expanding the training dataset to 2018 - 2025, introducing reward normalization, recalibrating the ΔQ threshold to 0.2, and replacing the retrain BRGap method with the greedy-q method. This setup satisfied the BRGap threshold at episode 2,501 for the multi-agent case and episode 2,251 for the single-agent case.

9.2 Market Implications

The market-level implications of the results were examined in Section 7.2. As shown in Table 12, the mean clearing price fell from \$37.49/MWh in the single-agent case to \$32.65/MWh in the three-agent case (-12.9%) and \$31.84/MWh in the five-agent case (-15.1%), as more

RL agents filled competitive dispatch positions in the merit order. We draw parallels to the deregulation of the Ontario electricity market in 2002 and the England and Wales wholesale electricity market in the late 1990s. It is shown that even when a market is opened up to more competitors, dominant participants in the market can continue to use bid markups and earn a disproportionate share of the money. In this project, the behaviour of the agents most closely resembled what happened in the England and Wales electricity market.

9.3 Engineering Ethics

As detailed in Section 7.3, RL-based bidding can create an information advantage. Participants with greater data resources and computing power can learn more adequate bidding patterns which can increase their market power. In addition, when many players are present, the market may end up with low bidding prices because of the overall competitive structure of the system and interaction of bids rather than deliberate actions from a single participant. This section also highlights the fact that tabular Q-learning allows us to directly analyze each action-state pair, making the learned strategy easier to interpret.

9.4 Limitations

Some of the limitations, detailed in Section 7.4, include the loss of information due to discretizing the state and action spaces into bins, modeling opponent bids using Gaussian distributions, a simplified hourly clearing process through a uniform-price scheme without the security constraints, and the use of Henry Hub spot prices as an estimate for the marginal cost without incorporating individual plant-level differences between opponents.

9.5 Future Work

In terms of future work that was outlined in Section 8, included are better modeling of opponents using opponent-class conditioning, increased complexity of both state and action spaces, and the move towards function approximation using deep Q-networks. Further features that could be introduced are the presence of ramping requirements, penalties for non-settlement, and considerations around reliability-based operational policies. Additionally, it is suggested that successful bidding algorithms be evaluated in terms of their fairness, market power, and regulatory compliance.

References

[1] Rishabh Agarwal et al. "Deep Reinforcement Learning at the Edge of the Statistical Precipice". In: *Advances in Neural Information Processing Systems*. 2021.

[2] Gurdal Arslan and Serdar Yüksel. "Decentralized Q-Learning for Stochastic Teams and Games". In: *IEEE Transactions on Automatic Control* 62.4 (Apr. 2017). Accessed February 23rd, 2026, pp. 1545–1558. DOI: 10.1109/TAC.2016.2598476.

[3] Clean Energy Wire. *Setting the Power Price: The Merit Order Effect*. Clean Energy Wire. 2015. URL: https://www.cleanenergywire.org/factsheets/setting-power-price-merit-order-effect (visited on 10/23/2025).

[4] Electric Reliability Council of Texas (ERCOT). *ERCOT Data Access Portal*. ERCOT. 2026. URL: https://data.ercot.com/ (visited on 03/20/2026).

[5] Electric Reliability Council of Texas (ERCOT). *ERCOT Nodal Protocols*. ERCOT. 2024. URL: https://www.ercot.com/mktrules/nprotocols/current (visited on 04/10/2026).

[6] Electric Reliability Council of Texas (ERCOT). *System-Wide Prices*. ERCOT. 2026. URL: https://www.ercot.com/gridmktinfo/dashboards/systemwideprices (visited on 02/20/2026).

[7] Eugene F. Fama and Kenneth R. French. "The Capital Asset Pricing Model: Theory and Evidence". In: *Journal of Economic Perspectives* 18.3 (2004), pp. 25–46. DOI: 10.1257/0895330042162430. URL: https://pubs.aeaweb.org/doi/10.1257/0895330042162430.

[8] Federal Energy Regulatory Commission. *Anti-Manipulation Rule*. Tech. rep. Order No. 670, 18 C.F.R. Part 1c. Federal Energy Regulatory Commission (FERC), Jan. 2006. URL: https://www.ferc.gov/enforcement-legal/enforcement/prohibition-energy-market-manipulation.

[9] Federal Energy Regulatory Commission. *Uplift in RTO and ISO Markets*. Staff White Paper. Federal Energy Regulatory Commission, Aug. 2014. URL: https://www.ferc.gov/sites/default/files/2020-05/08-13-14-uplift_3.pdf.

[10] Federal Energy Regulatory Commission. "Wholesale Electricity Markets – Chapter 2". In: A Handbook for Energy Market Basics: Energy Markets Primer. Ed. by Federal Energy Regulatory Commission. Accessed on April 13, 2026. 2024. Chap. 2. URL: https://www.ferc.gov/sites/default/files/2024-01/24_Energy-Markets-Primer_0117_DIGITAL_0.pdf.

[11] Ali Hortaçsu and Steven L. Puller. "Understanding Strategic Bidding in Multi-Unit Auctions: A Case Study of the Texas Electricity Spot Market". In: *The RAND Journal of Economics* 39.1 (2008), pp. 86–114. DOI: 10.1111/j.0741-6261.2008.00005.x.

[12] Junling Hu and Michael P. Wellman. "Nash Q-Learning for General-Sum Stochastic Games". In: *Journal of Machine Learning Research* 4 (Nov. 2003), pp. 1039–1069. URL: https://www.jmlr.org/papers/volume4/hu03a/hu03a.pdf (visited on 02/22/2026).

[13] International Energy Agency. *Electricity 2024: Analysis and Forecast to 2026*. Paris: IEA Publications, 2024. URL: https://www.iea.org/reports/electricity-2024.

[14] ISO New England. *How Resources Are Selected and Prices Are Set in the Wholesale Energy Markets*. Accessed: 2025-10-10. 2025. URL: https://www.iso-ne.com/about/what-we-do/how-resources-are-selected-and-prices-are-set.

[15] David S. Leslie and Edmund J. Collins. "Individual Q-Learning in Normal Form Games". In: *SIAM Journal on Control and Optimization* 44.2 (2005), pp. 495–514.

[16] Volodymyr Mnih et al. "Human-Level Control through Deep Reinforcement Learning". In: *Nature* 518.7540 (2015), pp. 529–533. DOI: 10.1038/nature14236. URL: https://www.nature.com/articles/nature14236.

[17] Costas Mylonas. *From an Oligopolistic Market to Socially Equitable Energy: Institutional Reforms within the Target Model Framework*. ENA Institute for Alternative Policies. Feb. 2026. URL: https://enainstitute.org/en/publication/from-an-oligopolistic-market-to-socially-equitable-energy-institutional-reforms-within-the-target-model-framework/ (visited on 04/05/2026).

[18] Potomac Economics. *2023 State of the Market Report for the ERCOT Electricity Markets*. Tech. rep. Potomac Economics, Ltd. (Independent Market Monitor for ERCOT), 2024. URL: https://www.potomaceconomics.com/wp-content/uploads/2024/05/2023-State-of-the-Market-Report_Final_060624.pdf.

[19] Navid Rashedi, Mohammad Amin Tajeddini, and Hamed Kebriaei. "Markov game approach for multi-agent competitive bidding strategies in electricity market". In: *IET Generation, Transmission & Distribution* 10.15 (2016), pp. 3756–3763. DOI: 10.1049/iet-gtd.2016.0075. URL: https://ietresearch.onlinelibrary.wiley.com/doi/full/10.1049/iet-gtd.2016.0075.

[20] Fred C. Schweppe et al. *Spot Pricing of Electricity*. Norwell, MA: Kluwer Academic Publishers, 1988. ISBN: 978-0-89838-260-4.

[21] Anjali Sheffrin. *Empirical Evidence of Strategic Bidding in California ISO Real-Time Market*. Technical Report. Attachment B. Prepared in support of CAISO filing. Covers bidding activity of 5 large in-state suppliers and 16 importers in the CA ISO real-time market, May–November 2000. California Independent System Operator (CAISO), Department of Market Analysis, Mar. 2001. URL: https://www.caiso.com/documents/attachmentb-empiricalevidence-strategicbiddingincaliforniaisorealtimemarket.pdf (visited on 04/01/2026).

[22] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. 2nd ed. Cambridge, MA: MIT Press, 2018.

[23] Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. 2nd ed. MIT Press, 2018.

[24] Andrew Sweeting. "Market Power in the England and Wales Wholesale Electricity Market 1995–2000". In: *The Economic Journal* 117.520 (2007), pp. 654–685. DOI: 10.1111/j.1468-0297.2007.02045.x.

[25] Athina Tellidou and Anastasios Bakirtzis. *A Q-learning agent-based model for the analysis of the power market dynamics*. Accessed: 2023-10-10. Available at https://inis.iaea.org/records/27ey5-qmn75. 2006.

[26] Michel Tokic and Gunnar Palm. "Value-Difference Based Exploration: Adaptive Control between Epsilon-Greedy and Softmax". In: *KI 2011: Advances in Artificial Intelligence*. Springer, 2011, pp. 335–346. DOI: 10.1007/978-3-642-24455-1_33.

[27] Michael J. Trebilcock and Roy Hrab. "Electricity Restructuring in Ontario". In: *The Energy Journal* 26.1 (2005), pp. 123–146. DOI: 10.5547/ISSN0195-6574-EJ-Vol26-No1-6.

[28] U.S. Energy Information Administration. *Sources of Price Volatility in the ERCOT Market*. STEO Supplement. U.S. Energy Information Administration, Oct. 2022. URL: https://www.eia.gov/outlooks/steo/special/supplements/2022/2022_sp_03.pdf.

[29] U.S. Energy Information Administration (EIA). *Annual Electric Power Industry Report, Form EIA-860 Detailed Data with Previous Form Data (EIA-860A/860B)*. Sept. 2025. URL: https://www.eia.gov/electricity/data/eia860/ (visited on 03/15/2026).

[30] Christopher J. C. H. Watkins and Peter Dayan. "Q-learning". In: *Machine Learning* 8.3-4 (1992), pp. 279–292. DOI: 10.1007/BF00992698. URL: https://link.springer.com/article/10.1007/BF00992698.

[31] Christopher J. C. H. Watkins and Peter Dayan. "Q-learning". In: *Machine Learning* 8.3–4 (1992), pp. 279–292.

[32] Brady Yauch, Brendan Callery, and Tyama Lyall. *MRP Week 3 Review: Day-ahead Market Failure and Ongoing Price Volatility in Real-Time*. Accessed: 2025-10-12. May 2025. URL: https://www.poweradvisoryllc.com/reports/mrp-week-3-review-day-ahead-market-failure-and-ongoing-price-volatility-in-real-time.

[33] Kazufumi Yuasa and Yoshiharu Takeuchi. "Optimized Energy Allocation Method Based on Capital Asset Pricing Model for Multi-use of Battery Energy Storage System". In: *2022 International Power Electronics Conference* (2022). DOI: 10.23919/IPEC-Himeji2022-ECCE53331.2022.9807094. URL: https://ieeexplore.ieee.org/document/9807094.

A Training and Simulation Pseudocode

Algorithm 1 `greedy_q` Best-Response Gap Estimation

Require: Agents $\{Q_i\}_{i=1}^{n}$, frozen policies $\{\pi^*_{-i}\}$, evaluation episodes K, discount factor γ
Ensure: $\text{BRGap}_{\text{norm}}$

for each agent $i = 1, \ldots, n$ **do**
  $J_{\text{curr}} \leftarrow 0; \quad J_{\text{BR}} \leftarrow 0$
  for $k = 1, \ldots, K$ **do**
    Current policy rollout: agent i plays softmax $\pi_i^*(\cdot \mid s; \tau)$, all $j \neq i$ play frozen π_j^*
    $J_{\text{curr}} \leftarrow J_{\text{curr}} + \sum_{t=0}^{T} \gamma^t r_{i,t}$
    Best-response rollout: agent i plays $\pi_i^{\text{BR}}(s) = \arg\max_a Q_i(s, a)$, all $j \neq i$ play frozen π_j^*
    $J_{\text{BR}} \leftarrow J_{\text{BR}} + \sum_{t=0}^{T} \gamma^t r_{i,t}$
  end for
  $\hat{J}_i \leftarrow J_{\text{curr}}/K; \quad \hat{J}_i^{\text{BR}} \leftarrow J_{\text{BR}}/K$
  $\text{BRGap}_i \leftarrow \max\big(0,\ \hat{J}_i^{\text{BR}} - \hat{J}_i\big)$
  $\text{BRGap}_i^{\text{norm}} \leftarrow \max\Big(0,\ \frac{\hat{J}_i^{\text{BR}} - \hat{J}_i}{|\hat{J}_i|}\Big)$
end for
return $\text{BRGap}_{\text{norm}} \leftarrow \dfrac{1}{n} \displaystyle\sum_{i=1}^{n} \text{BRGap}_i^{\text{norm}}$

B Hyperparameters and Experimental Settings

Number of agents	Warm Start: True	Warm Start: False
1	20.10 – 52.21	18.61 – 64.25
3	16.48 – 68.47	18.41 – 58.07
5	14.23 – 74.52	16.40 – 72.48

Table 8: Range of maximum agent ΔQ during the last 50 training episodes with and without warm start

Number of agents	Warm Start: True	Warm Start: False
1	7.79	8.51
3	14.82	12.61
5	14.84	15.12

Table 9: Std of maximum agent ΔQ during the last 50 training episodes with and without warm start

Parameter Name	**Type**	**Value Used - Single Agent**	**Value Used - Multi Agent**
brgap k eval	Threshold	20	20
brgap late save threshold	Threshold	0.25	0.25
brgap retrain episodes	Threshold	28	28
brgap train mode	Threshold	train	train
late save min episodes	Threshold	230	230
q delta threshold	Threshold	50	50
q delta window	Threshold	1	1
risk penalty λ	Risk	0.0	0.0
max notional q	Risk	0.95	0.95
warm start q	Learning	True	True
policy freeze k	Learning	5	10
learning rate	Learning	1.0	1.0
learning rate exponent	Learning	0.7	0.7
temperature	Temperature	1.0	1.0
temperature decay	Temperature	0.995	0.995
temperature min	Temperature	0.1	0.1
temperature mode	Temperature	exp decay	exp decay

Table 10: Table of model parameters and selected values for simulation.

Parameter Name	**Type**	**Value Used - 1- Agent**	**Value Used - 3-Agent**	**Value Used - 5-Agent**
brgap k eval	Threshold	60	60	60
brgap late save threshold	Threshold	0.10	0.25	0.25
brgap train mode	Threshold	greedy q	greedy q	greedy q
late save min episodes	Threshold	1500	1750	1750
q delta threshold	Threshold	0.2	0.2	0.2
q delta window	Threshold	10	10	10
risk penalty λ	Risk	0.0	0.0	0.0
max notional q	Risk	0.95	0.95	0.95
warm start q	Learning	True	True	True
policy freeze k	Learning	5	10	10
learning rate	Learning	1.0	1.0	1.0
learning rate exponent	Learning	0.7	0.7	0.7
temperature	Temperature	1.0	1.0	1.0
temperature decay	Temperature	0.9993	0.9993	0.9995*
temperature min	Temperature	0.05	0.05	0.05
temperature mode	Temperature	exp decay	exp decay	exp decay

*Slower decay rate used in the 5-agent case to sustain exploration over a larger joint action space.

Table 11: Table of model parameters and selected values for simulation on updated design.

C Additional Figures and Tables

C.1 Original Design

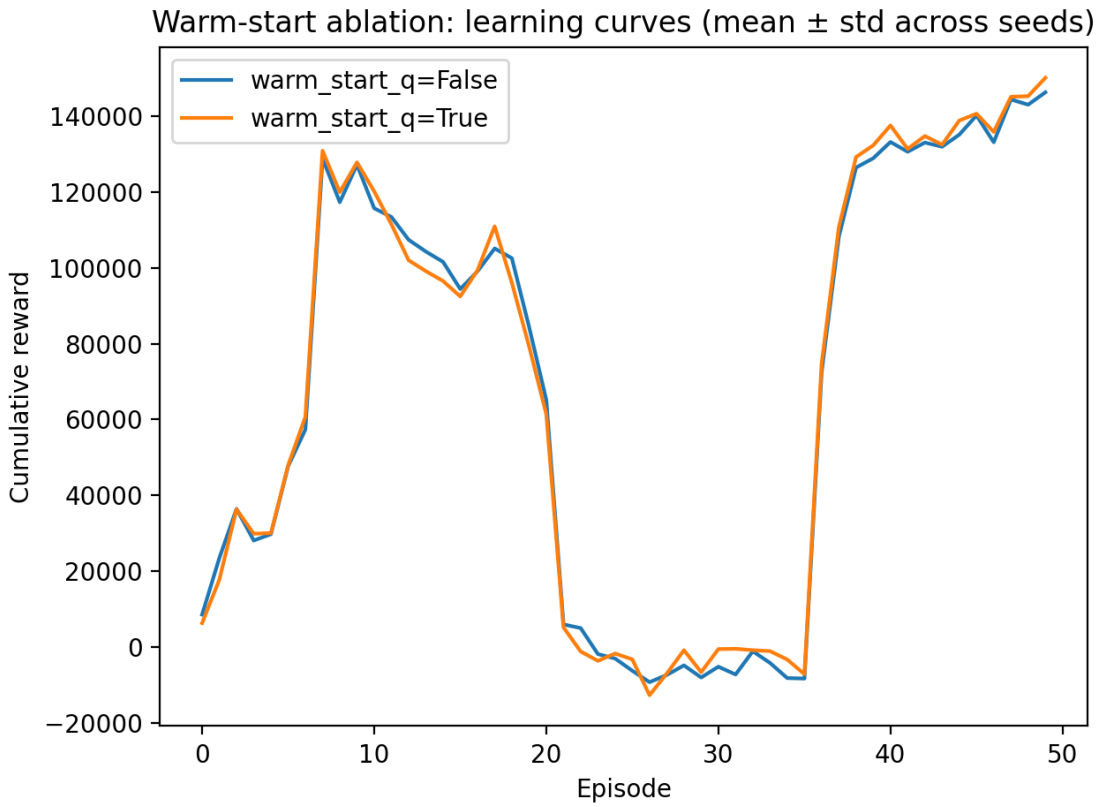


Figure 18: Warm Start ablation results on 50 episode sample size

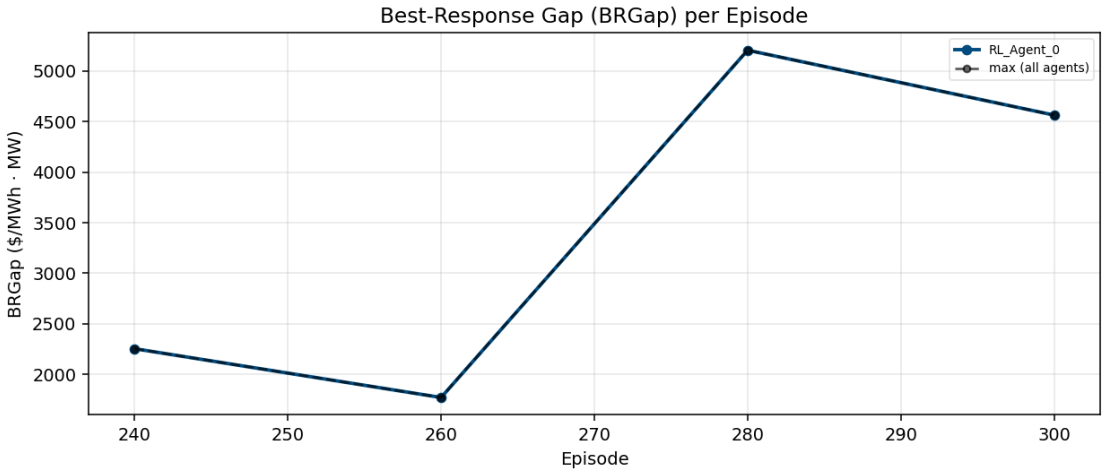


Figure 19: Normalized BRGap using `retrain` in single agent case.

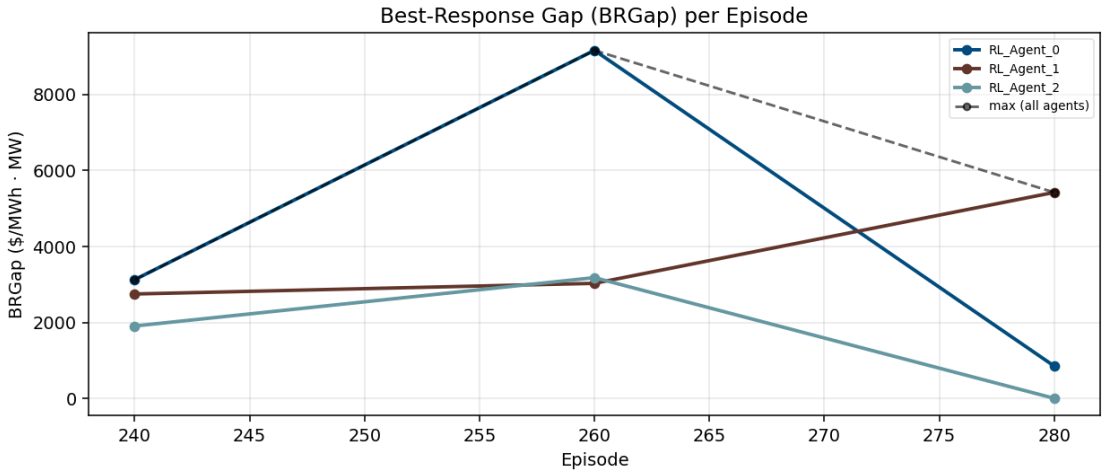


Figure 20: Normalized BRGap using `retrain` in three agent case.

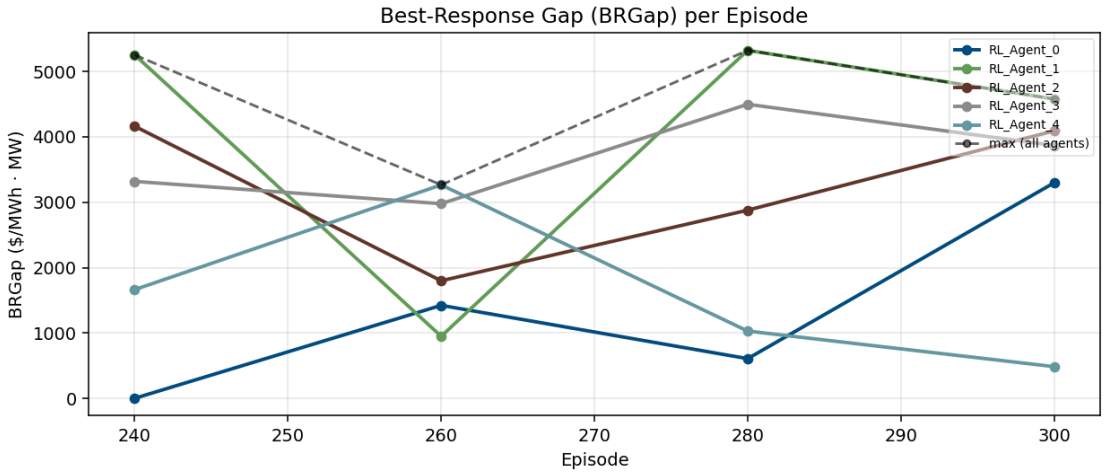


Figure 21: Normalized BRGap using `retrain` in five agent case.

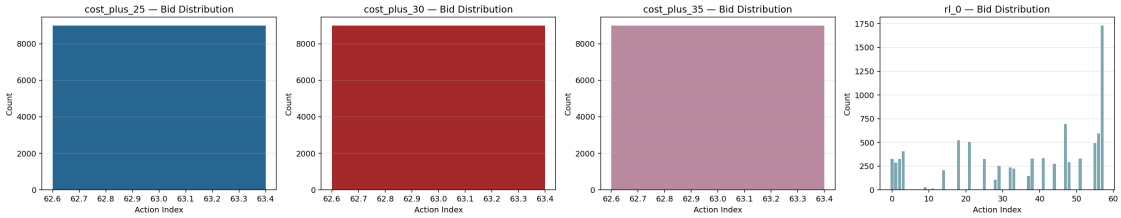


Figure 22: Bidding distributions in single-agent case compared to cost-plus markup baselines over the 2019 - 2024 ERCOT evaluation period.

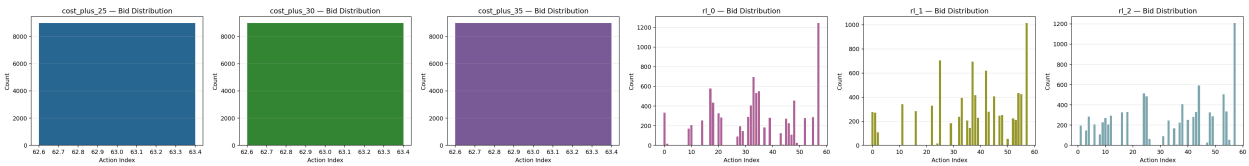


Figure 23: Bidding distributions in three-agent case compared to cost-plus markup baselines over the 2019-2024 ERCOT evaluation period.

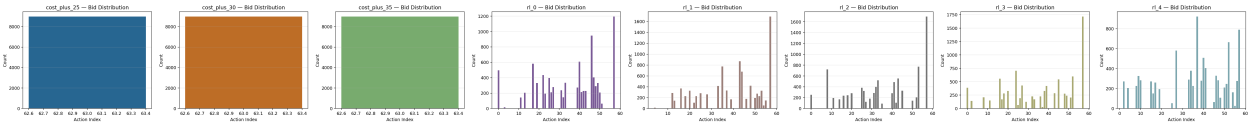


Figure 24: Bidding distributions in five-agent case compared to cost-plus markup baselines over the 2019-2024 ERCOT evaluation period.

Case	Mean RL Bid ($/MWh)	Mean RL Bid Efficiency (%)	Mean Clearing ($/MWh)	Δ_{clear}
1-agent	94.99	67.3 %	37.49	—
3-agent	62.29	67.6 %	32.65	−12.9%
5-agent	72.96	71.1 %	31.84	−15.1%

Table 12: Mean market clearing price and RL agent bid price, evaluated over the 2019-2024 ERCOT test period.

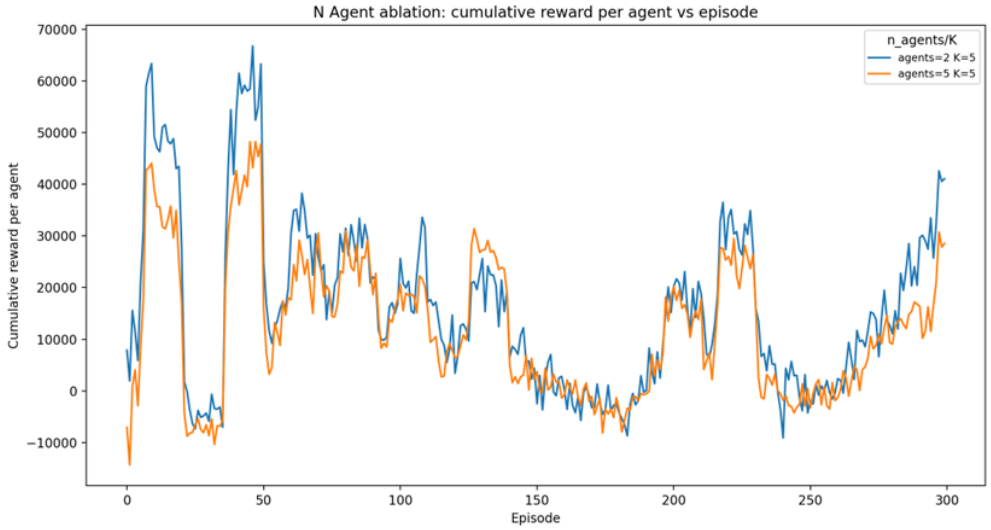


Figure 25: Cumulative reward across two and five agents.

C.2 Updated Design

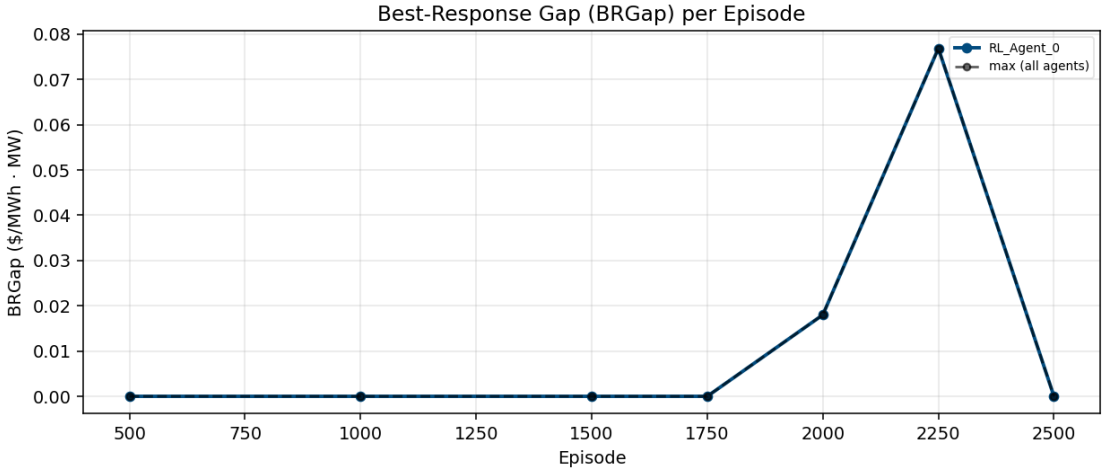


Figure 26: Normalized BRGap using `greedy q` in single agent case.

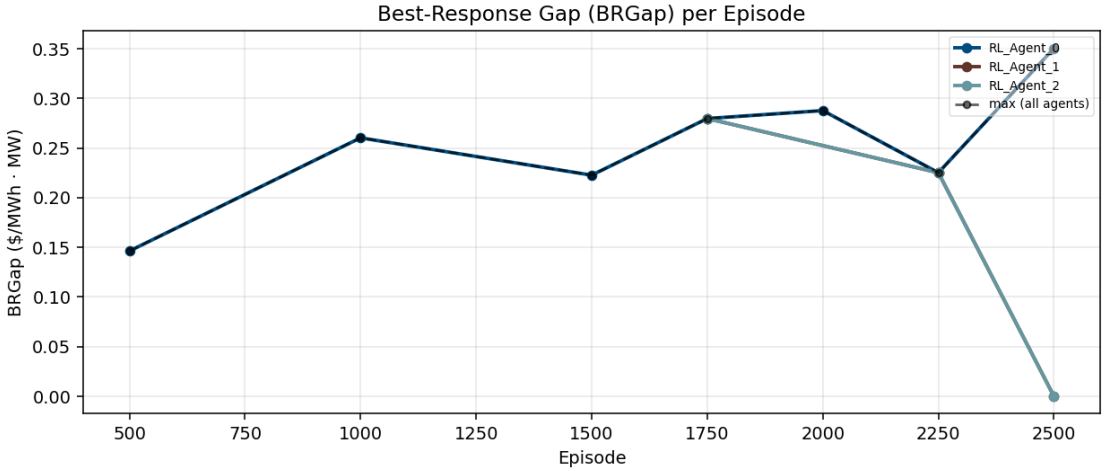


Figure 27: Normalized BRGap using `greedy q` in three agent case.

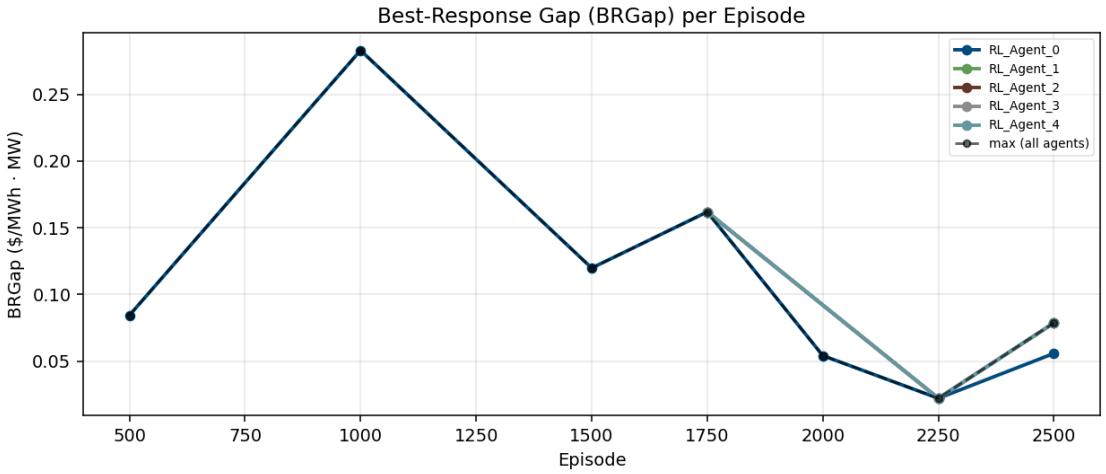


Figure 28: Normalized BRGap using `greedy q` in five agent case.

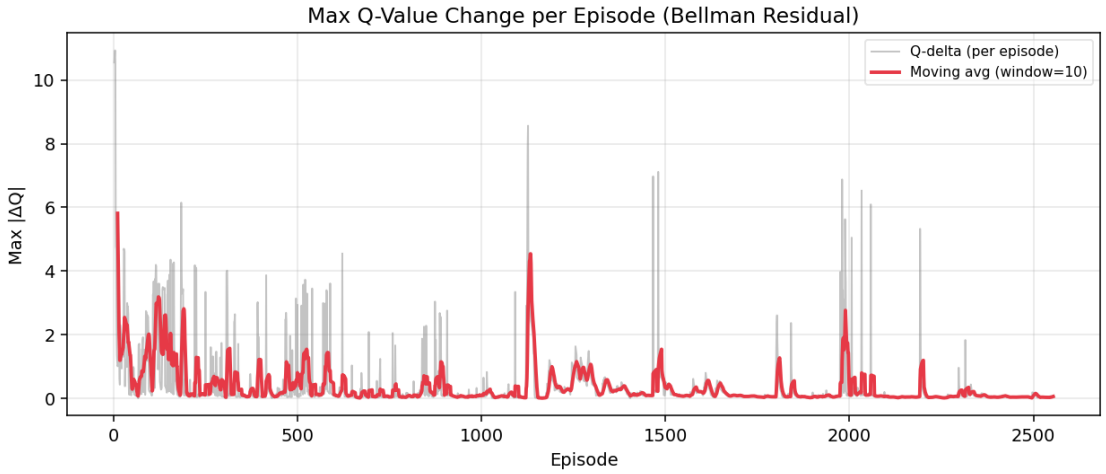


Figure 29: Normalized ΔQ across training in the extended single-agent case.

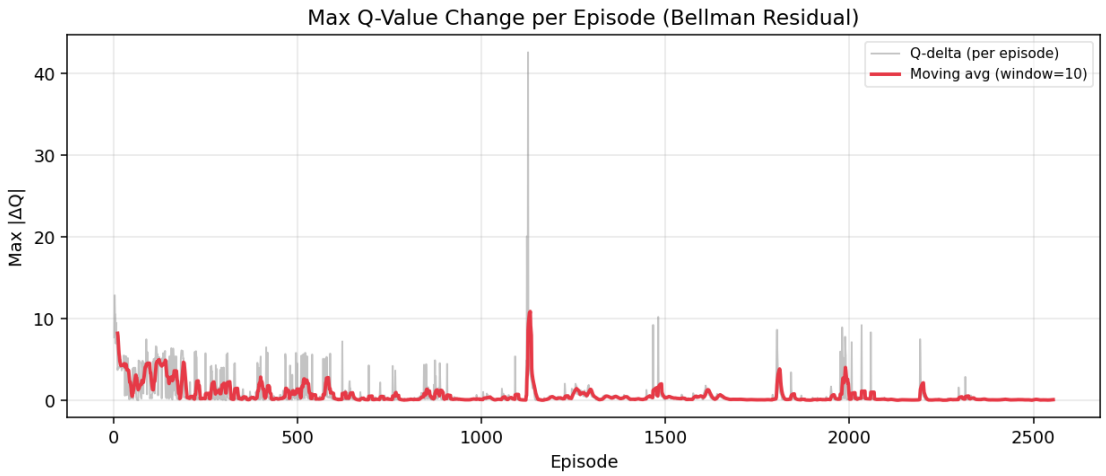


Figure 30: Normalized ΔQ across training in the extended three-agent case.

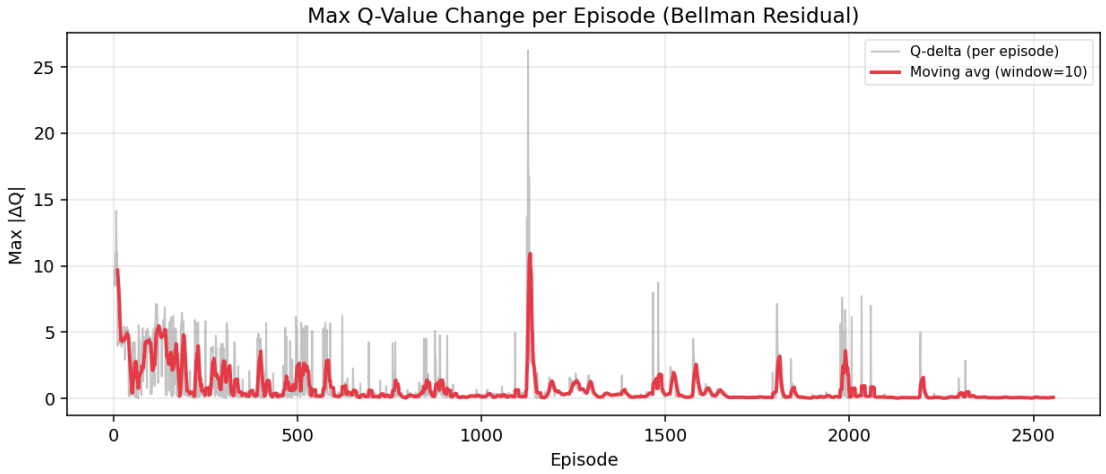


Figure 31: Normalized ΔQ across training in the extended five-agent case.

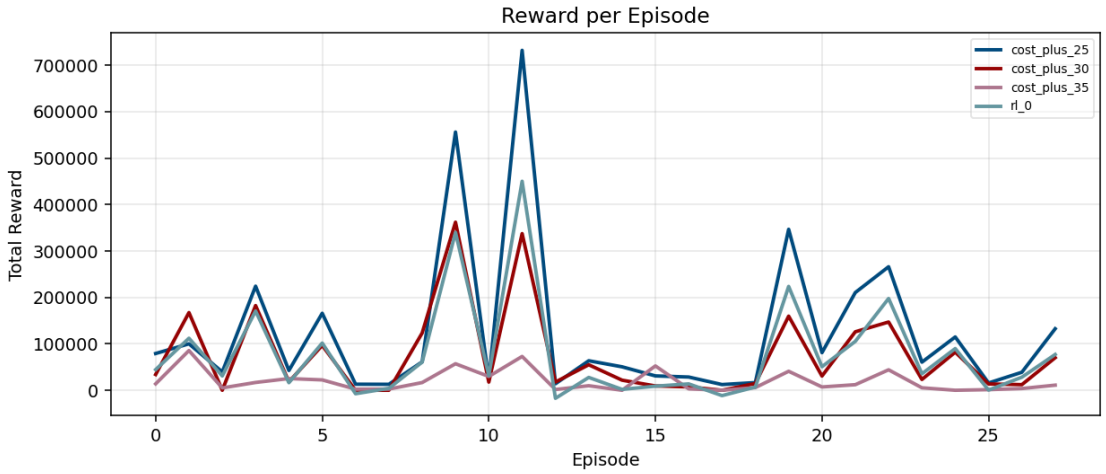


Figure 32: Episode profits in the extended single-agent case vs. cost-plus baselines, in-sample 2019 - 2024 ERCOT evaluation.

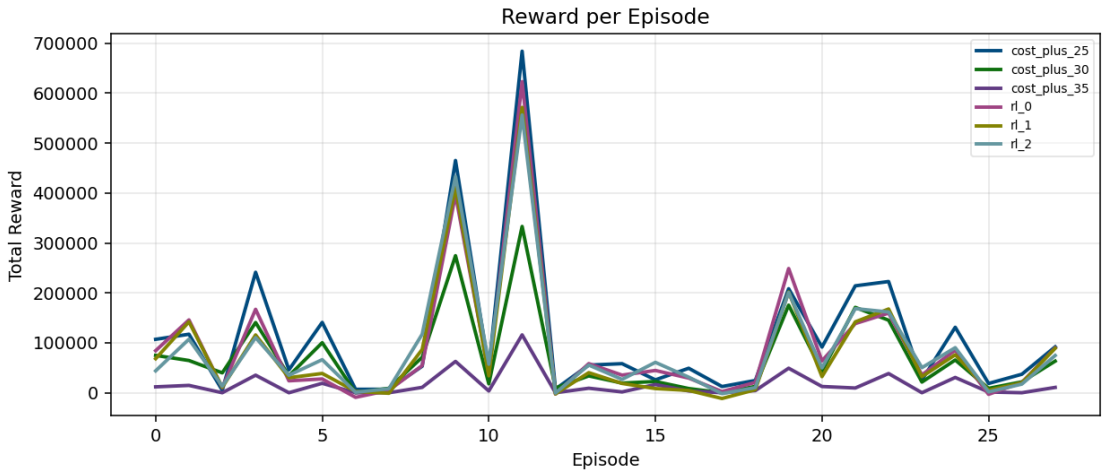


Figure 33: Episode profits in the extended three-agent case vs. cost-plus baselines.

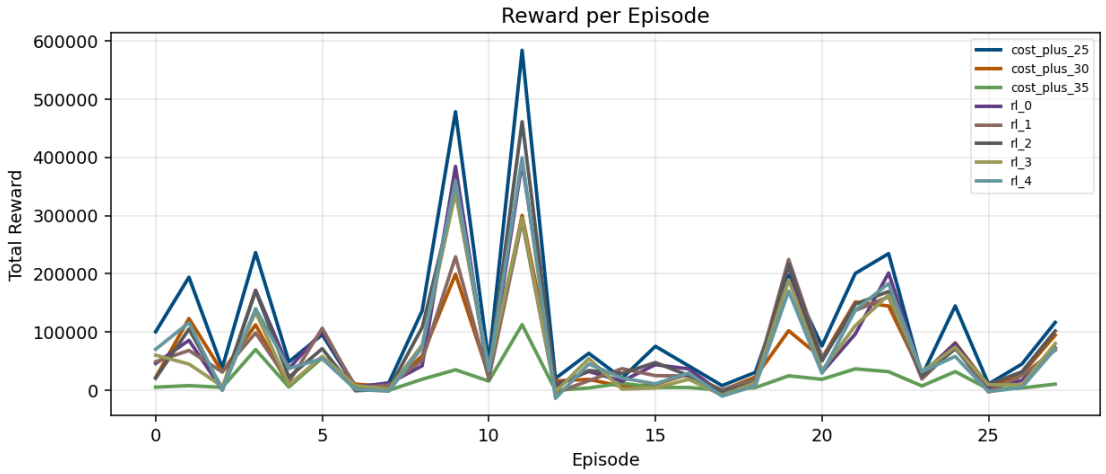


Figure 34: Episode profits in the extended five-agent case vs. cost-plus baselines.

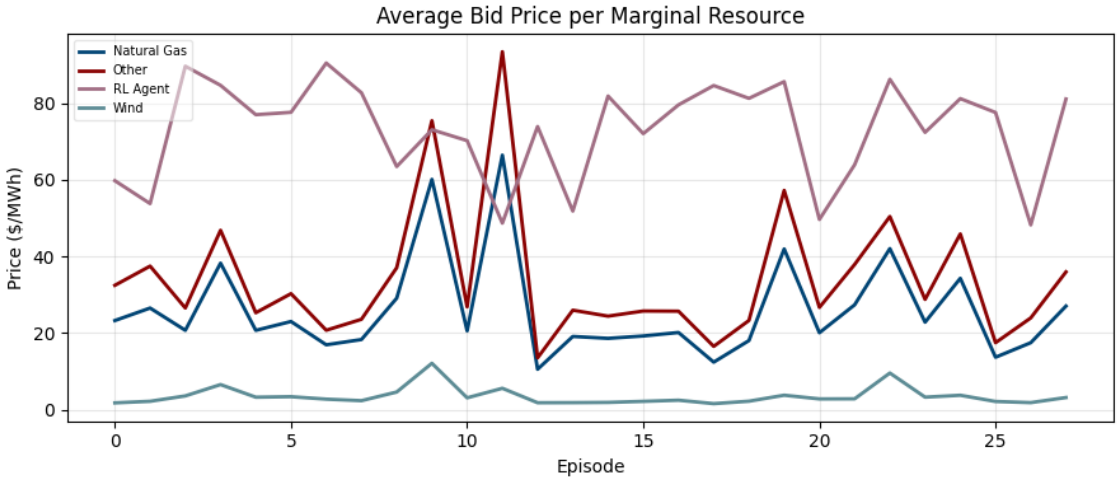


Figure 35: Average Bid Price per Marginal resource in 5 agent baseline case.

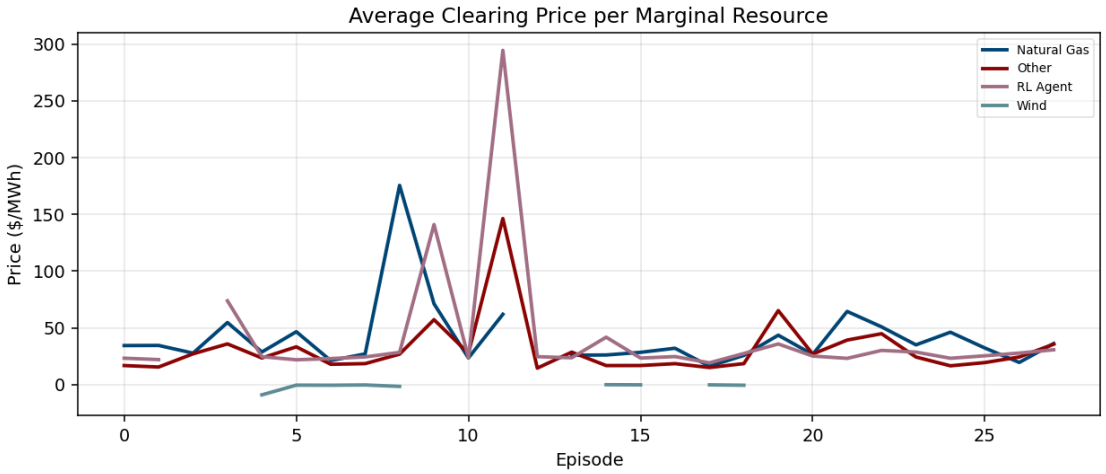


Figure 36: Clearing Price per Marginal resource in 5 agent baseline case.

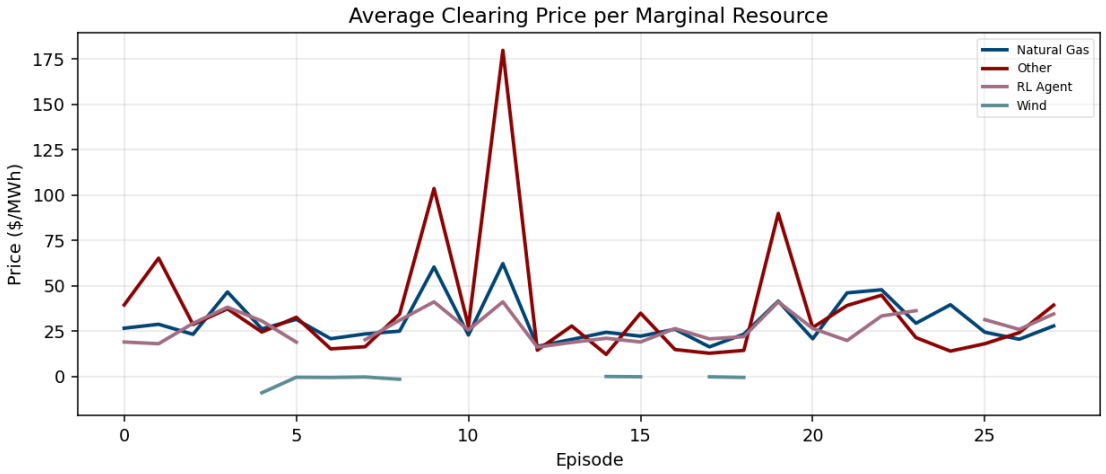
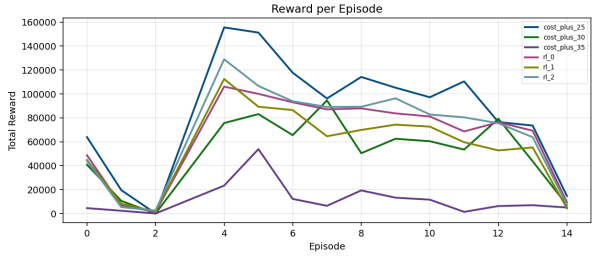


Figure 37: Clearing Price per Marginal resource in 5 agent baseline case within iterated design.

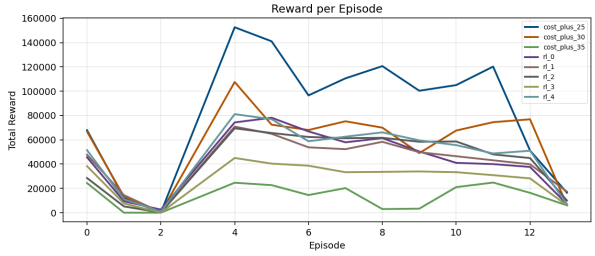
(a) Single-agent (baseline vs. extended, Dec 2025 episodes).

(b) Single-agent (2018-2025 training comparison).

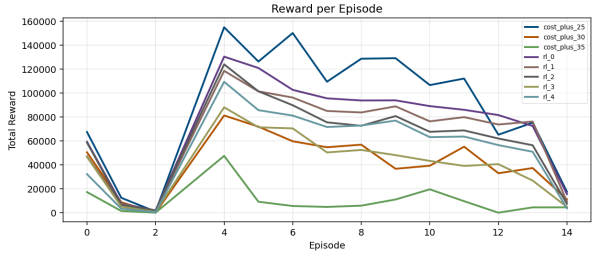
(c) Three-agent (baseline vs. extended, Dec 2025 episodes).



(d) Three-agent (2018-2025 training comparison).



(e) Five-agent (baseline vs. extended, Dec 2025 episodes).



(f) Five-agent (2018-2025 training comparison).

Figure 38: Sequential comparison of RL agent profit for the original and iterated designs and across the different agent-count configuration.

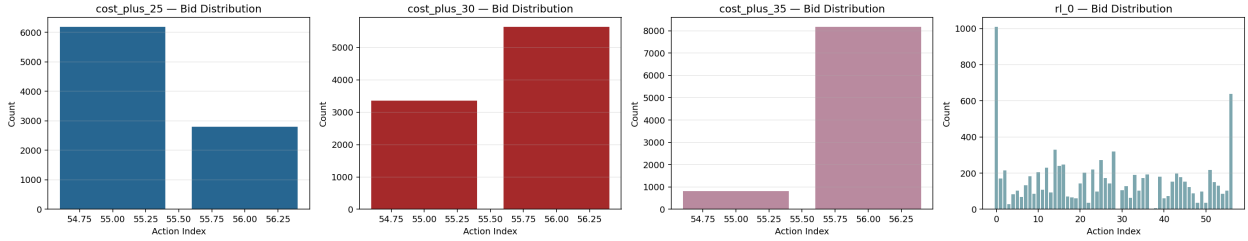


Figure 39: Bidding distributions in single-agent case compared to cost-plus markup baselines over the 2019 - 2024 ERCOT evaluation period.

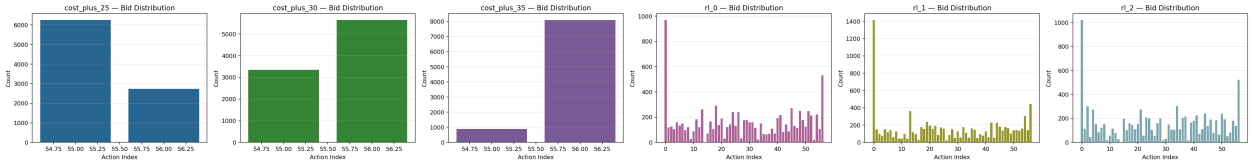


Figure 40: Bidding distributions in three-agent case compared to cost-plus markup baselines over the 2019-2024 ERCOT evaluation period.

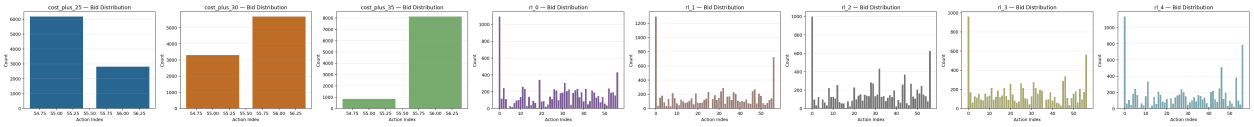


Figure 41: Bidding distributions in five-agent case compared to cost-plus markup baselines over the 2019-2024 ERCOT evaluation period.

Agents	Range (min, max)	Std	Mean
1	[0.0189, 0.1227]	0.0268	0.0450
3	[0.0252, 0.1315]	0.0324	0.0580
5	[0.0265, 0.1048]	0.0247	0.0538

Table 13: ΔQ statistics over the final 50 training episodes for updated runs. All values below the `q delta threshold` = 0.20 stopping criterion, confirming convergence.

Case	Mean ΔQ (first 25 ep.)	Max ΔQ (first 25 ep.)
1- agent	3.189	10.934
3- agent	5.825	12.859
5- agent	6.457	14.148

Table 14: Mean and peak ΔQ during the first 25 training episodes for updated runs.

Agent	Mean Profit ($)	Profit Std ($)	Win Rate (%)	Profit Share (%)	Mean Bid ($/MWh)	Min Profit ($)	CVaR_{10} ($)
Cost+25%	126,146	170,958	82.1	101.80	279.15	12,317	12,582
Cost+30%	76,036	96,483	10.7	24.75	523.03	–	–
Cost+35%	19,581	23,443	3.6	9.88	739.24	–	–
RL_0	78,355	110,032	3.6	−36.42	73.73	−17,341	−14,422

Table 15: Evaluation of updated desing in Single agent case vs. cost-plus markup; in-sample evaluation (2019-2024, $k = 28$ episodes).

Agent	Mean Profit ($)	Profit Std ($)	Win Rate (%)	Profit Share (%)	Mean Bid ($/MWh)	Min Profit ($)	$CVaR_{10}$ ($)
Cost+25%	114,069	151,020	64.3	24.36	275.64	4,867	5,877
Cost+30%	70,724	83,552	10.7	15.10	523.99	–	–
Cost+35%	16,440	25,420	0.0	3.51	732.71	–	–
RL_0	92,583	136,542	10.7	19.77	64.90	−9,225	−6,218
RL_1	83,580	130,115	0.0	17.85	54.80	−11,553	−6,978
RL_2	90,901	127,175	14.3	19.41	62.59	−1,791	−1,238

Table 16: Evaluation of updated design in Three agent case vs. cost-plus markup; in-sample evaluation (2019-2024, $k = 28$ episodes).

Agent	Mean Profit ($)	Profit Std ($)	Win Rate (%)	Profit Share (%)	Mean Bid ($/MWh)	Min Profit ($)	$CVaR_{10}$ ($)
Cost+25%	117,580	137,681	82.1	20.75	280.57	4,680	6,381
Cost+30%	62,407	70,584	3.6	11.01	528.16	–	3,088
Cost+35%	19,215	25,291	0.0	3.39	734.96	–	–
RL_0	78,516	104,739	0.0	13.85	56.44	−4,576	−1,963
RL_1	66,239	76,866	10.7	11.69	80.46	−4,672	−3,876
RL_2	82,727	108,894	0.0	14.60	72.51	−2,176	−1,555
RL_3	66,507	87,600	3.6	11.73	65.59	−8,407	−5,586
RL_4	73,577	102,250	0.0	12.98	86.68	−13,770	−11,899

Table 17: Evaluation of updated design in Five agent case vs. cost-plus markup; in-sample evaluation (2019-2024, $k = 28$ episodes).

Case	Mean RL Bid ($/MWh)	Bid Efficiency (%)	Mean Clearing ($/MWh)	Δ Clearing (%)
1- agent	73.73	38.9	32.96	–
3- agent	60.77	41.9	31.69	−3.85 %
5- agent	72.33	45.2	31.43	−4.64%

Table 18: RL agent bidding and market-clearing statistics for updated runs, in-sample evaluation (2019-2024, $k = 28$ episodes).

C.3 Sequential Comparison: Original vs. Iterated design

Case	Agents	Mean Profit ($)	Profit Std ($)	Win Rate (%)	Profit Share (%)	Mean Bid ($/MWh)	Min Profit ($)	$CVaR_{10}$ ($)
Original Design, 1 agent	Cost+25%	81,862	52,812	76.9	39.89	283.27	–	–
	Cost+30%	55,483	39,288	15.4	27.04	445.70	–	–
	Cost+35%	14,549	11,689	0.0	7.09	724.05	–	–
	RL_0	53,312	35,125	7.7	25.98	192.41	–	–
Original Design, 3 agents	Cost+25%	82,336	54,531	76.9	27.84	254.57	–	–
	Cost+30%	58,363	38,657	0.0	19.73	428.54	–	–
	Cost+35%	12,117	9,666	0.0	4.10	697.94	–	–
	RL_0	62,642	42,146	15.4	21.18	148.35	–	–
	RL_1	34,284	20,657	0.0	11.59	102.32	–	–
	RL_2	46,003	27,790	7.7	15.56	105.35	–	–
Original Design, 5 agents	Cost+25%	78,201	53,243	69.2	23.16	237.35	–	–
	Cost+30%	53,376	34,197	7.7	15.81	431.50	–	–
	Cost+35%	12,924	10,297	0.0	3.83	723.24	–	–
	RL_0	40,868	26,740	7.7	12.10	114.90	–	–
	RL_1	39,944	22,864	15.4	11.83	187.98	–	–
	RL_2	40,994	26,510	0.0	12.14	180.23	–	–
	RL_3	26,372	15,754	0.0	7.81	134.35	–	–
	RL_4	45,023	28,115	0.0	13.33	122.99	–	–
Updated Design, 1 agent	Cost+25%	90,107	58,358	100.0	44.14	301.66	–	–
	Cost+30%	49,993	33,039	0.0	24.49	538.61	–	–
	Cost+35%	12,032	12,422	0.0	5.89	738.08	–	–
	RL_0	52,008	36,084	0.0	25.48	67.27	−2,730	−2,730
Updated Design, 3 agents	Cost+25%	79,706	51,082	85.7	25.07	278.17	–	–
	Cost+30%	48,446	31,169	7.1	15.24	518.87	–	–
	Cost+35%	11,080	13,653	0.0	3.49	725.95	–	–
	RL_0	61,274	38,065	0.0	19.27	53.06	–	–
	RL_1	53,049	35,030	0.0	16.69	38.63	–	–
	RL_2	64,362	42,191	7.1	20.24	76.24	–	–
Updated Design, 5 agents	Cost+25%	83,734	54,376	78.6	20.19	266.25	–	–
	Cost+30%	39,735	25,392	0.0	9.58	549.24	–	–
	Cost+35%	9,370	12,075	0.0	2.26	745.85	–	–
	RL_0	70,039	43,398	7.1	16.89	52.56	–	–
	RL_1	62,982	39,665	7.1	15.19	113.19	–	–
	RL_2	58,217	38,010	7.1	14.04	75.21	–	–
	RL_3	39,185	27,216	0.0	9.45	53.04	–	–
	RL_4	51,419	35,318	0.0	12.40	124.35	–	–

Table 19: Statistical summary of sequential comparison of RL agent profit on each case between the original and updated design.

Case	Mean RL Bid ($/MWh)	Bid Efficiency (%)	Mean Clearing ($/MWh)	Δ Clearing (%)
Original Design				
1-agent	192.41	65.4	43.35	–
3-agent	118.67	60.9	35.21	−18.78
5-agent	148.09	65.0	34.43	−20.57
Iterative Design				
1-agent	67.27	35.2	37.67	–
3-agent	55.98	39.1	34.84	−7.52
5-agent	83.67	42.0	34.32	−8.90

Table 20: Mean RL bid price, bid efficiency, mean market clearing price, and percentage change in clearing price relative to the single-agent case, computed from sequential evaluation episodes.

C.4 Other

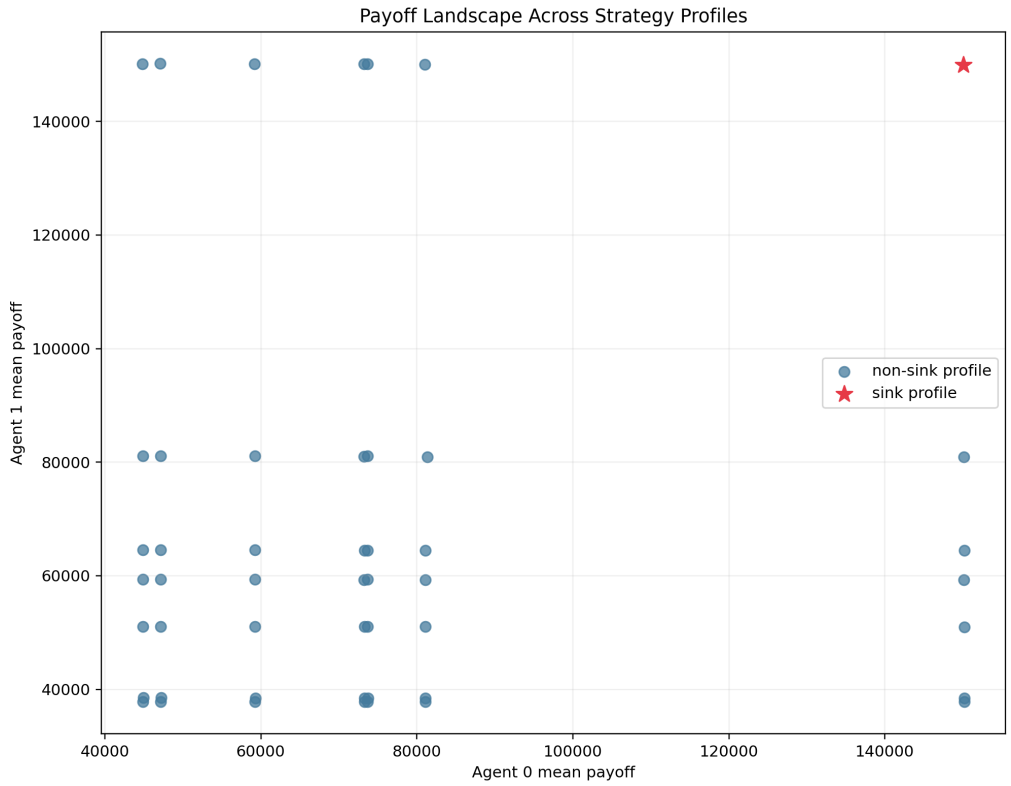


Figure 42: Sink profile convergence results from weak acyclicity verification.

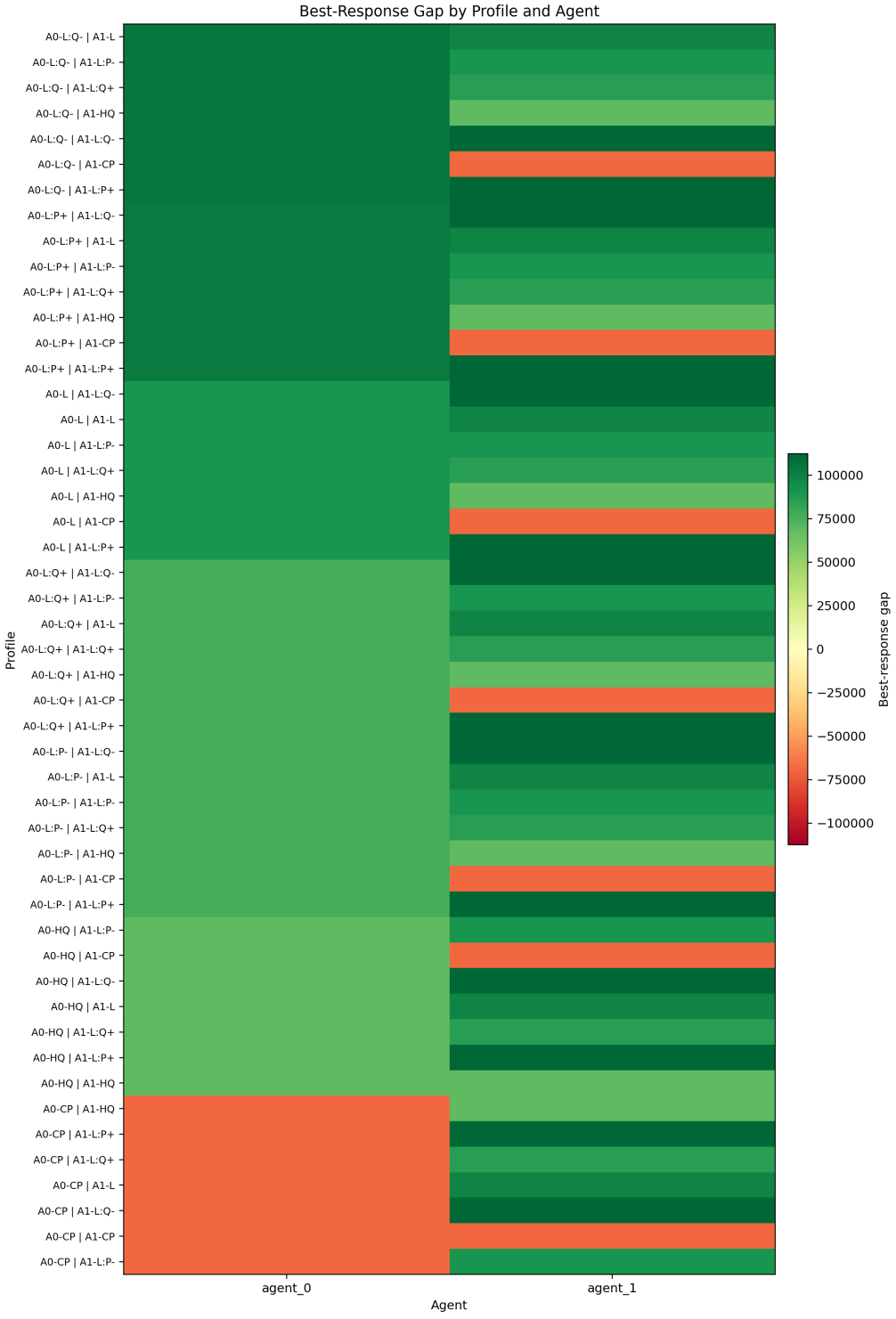


Figure 43: Best response gap results from weak acyclicity verification.

What We Assumed	Why We Assumed This	What This Breaks
Discretized state/action space	Allows for tabular Q-Learning	Reduced bidding options
Feasibility restrictions on actions	Enforces agent's make "safe" bids	Biases learning
Fuel-based opponent model	Scalable/interchangeable competitor simulation	Limited strategic realism
Uniform-price hourly clearing	Simpler market simulation	Reduces some of the intricacies of market
Opponent bids sampled from Gaussian distribution (within fuel band)	Needed stochastic way to generate heterogeneous bids	Shape of opponent bids is assumed, not inferred
Marginal cost modeled via Henry Hub gas price distribution	Data grounded proxy for natural gas marginal cost	Captures fuel cost variation, but ignores plant-specific costs

Figure 44: List of model Limitations