Digital Cartels? Simulating Hub-and-Spoke Collusion in Rental Markets via Shared Weight Reinforcement Learning

Rowan Morkner

2026-04-21

Abstract

When competing landlords subscribe to the same algorithmic pricing service, does the shared algorithm push rents above competitive levels? I build a multi-agent reinforcement learning simulation to find out. The model has 2,000 heterogeneous tenants with incomes drawn from ACS 2022 five-year microdata for Sacramento, CA, and landlords whose rents are set by a shared Proximal Policy Optimization (PPO) policy that reacts to each landlord's own local conditions. I run a full-factorial sweep across 144 configurations, varying the number of landlords (2–20), moving costs, excess supply, marginal cost, and tenant price sensitivity. Three findings stand out. First, the shared algorithm consistently learns prices above a competitive rule-based baseline, and the size of this premium shrinks steadily as the number of competitors goes up (a doubling of N cuts PPO prices by roughly 55%). Second, the housing-market frictions I sweep over (moving costs, supply tightness, demand elasticity) don't meaningfully change the concentration-price relationship: the friction interaction terms are small and mostly statistically insignificant, so the core result from simpler Bertrand-style models carries over to a richer environment. Third, and more tentatively, marginal-cost pass-through into prices is modest and mostly statistically insignificant (log-price-to-log-cost elasticities of roughly 0.08–0.22 across specifications), well below the near-unity pass-through one would expect from a textbook profit-maximizer. Part of this attenuation is mechanical (at $N = 2$, prices pin to the action-space cap regardless of cost), so this finding should be read as suggestive evidence that PPO pricing is driven more by market structure than by operating costs, rather than as a clean point estimate of true pass-through. The main contribution is showing that the basic algorithmic pricing result holds up in a housing market with search frictions, lease contracts, capacity

constraints, and heterogeneous tenants, not just in a stylized oligopoly setup.

1 Introduction

Algorithmic pricing tools are spreading quickly through rental housing markets, and this has caught the attention of both regulators and researchers. The U.S. Department of Justice recently filed an antitrust case against RealPage, a company whose YieldStar software gives rent recommendations to competing landlords using shared proprietary data (U.S. Department of Justice, 2024). The case raises a basic question: can a shared pricing algorithm push rents above competitive levels, even if the landlords using it never explicitly coordinate?

There is growing evidence that algorithms can learn to keep prices high on their own. Calvano et al. (2020) showed that even fully independent Q-learning agents in a simple price-competition game learn to sustain prices above competitive levels and punish deviations, behaving much like tacit colluders. When firms share a common algorithm, as in the RealPage case, the coordination channel is more direct: a single pricing policy, trained on pooled market data, is deployed across competing landlords, with each one receiving recommendations based on its own local conditions. Other work has tested whether algorithmic collusion holds up across different setups, market structures, and competitive conditions (Banchio & Skrzypacz, 2022; Johnson et al., 2023; Klein, 2021).

However, most of these studies use simplified market models (typically symmetric firms, identical products, zero switching costs, and frictionless demand) that leave out what makes housing markets different from a textbook oligopoly. Economic theory has a lot to say about when collusion works and when it doesn't: collusion is easier in concentrated markets with stable demand and visible prices, and harder when there are lots of frictions that make it hard to see or punish deviations (Ivaldi et al., 2003; Tirole, 1988). Rental housing has long-term leases that lock in prices, moving costs that give incumbent landlords an advantage, limited units per landlord, and tenants with very different incomes. These features could either help or hinder algorithmic collusion, and their combined effect is an open question that existing work has not tackled.

This paper asks: **How do market structure parameters, particularly the number of competing landlords and housing-specific frictions, shape the pricing equilibria that emerge when landlords use a common algorithmic pricing service?**

To answer this, I build a multi-agent reinforcement learning (MARL) simulation of a rental housing market. The simulation has three main pieces: 2,000 heterogeneous tenants whose incomes are drawn from Census microdata, a McFadden conditional logit tenant decision model (incorporating price sensitivity, incumbency bias, affordability constraints, and an outside option), and a shared-parameter PPO pricing policy deployed across all landlords. The shared policy represents a common algorithmic pricing service: one algorithm trained on pooled market experience that sets rents for each landlord based on local conditions, similar to how platforms like RealPage's YieldStar work in practice. I compare outcomes under the shared PPO policy against a competitive baseline (an occupancy-targeting rule-based agent) across 144 configurations varying five market structure parameters.

The main findings are as follows. First, PPO agents consistently learn prices above the competitive baseline at 2, 5, and 10 landlords, with the 20-landlord case converging close to the baseline. The size of this premium depends strongly on concentration: duopoly markets saturate at the simulator's administrative price cap of $10,000 (see the glossary in Appendix A for why this cap is different from an economic price ceiling), while twenty-landlord markets end up much closer to the competitive equilibrium. Quantitatively, the estimated elasticity of PPO prices with respect to the number of landlords is about -1.18, so a doubling of N cuts PPO prices by roughly 55%. Second, the housing-market frictions I sweep over don't meaningfully change this concentration-price relationship. The interaction effects between the number of landlords and each friction (moving cost, supply tightness, demand elasticity) are small and mostly not statistically significant, so the basic algorithmic pricing result from simple Bertrand-style oligopoly models carries over to a richer environment with little modification. In short, moving costs and supply tightness don't save renters from algorithmic markups, at least in this setup. Third, and more tentatively, marginal-cost pass-through into prices is modest and typically statistically insignificant, with estimated log-price-to-log-cost elasticities between roughly 0.08 and 0.22 across specifications. This is well below the near-unity pass-through a textbook profit-maximizer would deliver, suggesting that PPO agents price primarily off market structure rather than off operating costs. This finding is partly mechanical: in the $N = 2$ configurations PPO prices saturate at the simulator's action-space cap regardless of marginal cost, which deflates the estimated pass-through. It should therefore be read as suggestive evidence about PPO behavior rather than as a clean point estimate.

This paper contributes to the algorithmic-collusion literature in three ways. First, the shared-parameter PPO architecture is analogous to the hub-and-spoke structure of real-world pricing platforms, where a single vendor (the hub) trains one algorithm and deploys it across competing firms (the spokes). Second, by showing that the concentration-price relationship looks basically the same in a rich housing market as it does in simple Bertrand models, the results serve as a robustness check on prior work: the basic finding isn't just an artifact of the stylized assumptions in those models. Third, this is the first systematic parameter sweep I know of that looks at how market structure interacts with a shared algorithmic pricing service in a housing setting.

The remainder of the paper is organized as follows. Section 2 reviews the relevant literature. Section 3 describes the simulation model in detail. Section 4 outlines the experimental design. Section 5 presents the results. Section 6 discusses implications, limitations, and directions for future work. Section 7 concludes.

2 Literature Review

Research on algorithmic pricing and collusion has grown quickly since Calvano et al. (2020) showed that Q-learning agents can learn to keep prices above competitive levels in a Bertrand oligopoly. This section covers four areas: algorithmic collusion theory, real-world evidence on algorithmic pricing, multi-agent reinforcement learning in economic settings, and computational housing market models.

2.1 Algorithmic Collusion

Before getting into the algorithmic side, it's worth noting that economists have long known firms can sustain prices above competitive levels without any explicit agreement. In a repeated pricing game, if firms expect to keep running into each other, they can hold prices high by threatening to punish anyone who undercuts (Tirole, 1988). This kind of collusion works best when there are few competitors, prices are easy to see, and demand is steady. It tends to fall apart when markets are crowded, prices are opaque, or demand jumps around a lot (Ivaldi et al., 2003). Algorithmic pricing is a new context for an old problem.

The idea that algorithms might learn to collude without talking to each other goes back to at least Ivaldi et al. (2003), but actual computational evidence only came with Calvano et al. (2020). They trained tabular Q-learning agents in a simple Bertrand pricing game with logit demand and found that two to four symmetric firms consistently settle on prices above the Nash equilibrium, close to the joint monopoly level. The agents also develop punishment strategies: if one agent cuts its price, the others retaliate with temporary price wars before returning to the high-price outcome. None of this requires any explicit coordination.

An important follow-up question is whether these results only hold for Q-learning or whether other algorithms produce collusion too. Hettich (2021) compared DQN and PPO in algorithmic pricing and found that PPO converges faster and reaches higher collusion levels (collusion index of 0.43 vs. DQN's 0.23). Collusion did weaken, though, in larger markets with up to 10 firms. Friedrich et al. (2025) extended collusion research to episodic markets with inventory constraints and perishable goods, using DQN and PPO in duopoly settings with multinomial logit demand. Their PPO agents first converged to competitive prices, then gradually drifted upward toward collusion. This work is the closest to the present paper in terms of methodology: it uses the same algorithms, a similar demand model, and similar capacity constraints, though it studies perishable goods rather than housing. Schlechtinger et al. (2024) tested RL agents across a range of environment conditions and found that they reliably converge to inflated pricing regardless of the demand model, which supports the idea that collusion findings are not just artifacts of a particular setup.

2.2 Empirical Evidence on Algorithmic Pricing

Assad et al. (2024) offer the strongest real-world evidence that algorithmic pricing leads to higher prices. They studied the German retail gasoline market and found that margins rose 28% in duopoly markets when both stations adopted algorithmic pricing software. No price effect showed up in local monopolies, and margins only increased when all firms in a market adopted the software. This pattern is consistent with tacit collusion rather than each firm simply optimizing on its own.

In housing, the most notable case involves RealPage, whose YieldStar algorithm recommends rents to multifamily landlords using proprietary data pooled across competing properties (Vogell, 2022). The U.S. Department of Justice filed an antitrust complaint alleging that RealPage's data-sharing arrangement was an illegal hub-and-spoke conspiracy (U.S. Department of Justice, 2024). Calder-Wang & Kim (2024) provide the key empirical analysis. They found that buildings using the same pricing algorithm set more coordinated prices, and this coordination channel produced an average markup increase of $25/unit/month across roughly 4.2 million units. The White House Council of Economic Advisers estimated that coordinated algorithmic rents cost renters $70/month on average, totaling $3.8 billion nationally (White House Council of Economic Advisers, 2024).

On the legal side, Harrington (2018) argues that collusion by autonomous pricing software does not currently violate Section 1 of the Sherman Act (which requires an agreement between parties) and proposes ways to make autonomous algorithmic collusion illegal. Ezrachi & Stucke (2016) identify four scenarios for algorithmic collusion: Messenger, Hub-and-Spoke, Predictable Agent, and Digital Eye. This paper's shared-parameter IPPO setup maps most closely to the Hub-and-Spoke scenario. A common algorithm acts as the coordinating hub, and competing landlords are the spokes, each getting pricing recommendations based on local market conditions. This mirrors how platforms like RealPage work in practice, where one vendor's algorithm sets rents across competing properties.

2.3 Multi-Agent Reinforcement Learning

This paper builds on the multi-agent reinforcement learning (MARL) literature. Schulman et al. (2017) introduced PPO, the algorithm used here, which is designed to learn stable policies without requiring complex optimization procedures. Yu et al. (2022) showed that independent PPO (IPPO) performs surprisingly well across multi-agent benchmarks, matching or beating specialized methods. This directly supports the IPPO approach taken in this paper.

2.4 Computational Housing Market Models

The tenant decision model in this paper is built on the McFadden conditional logit framework, which is the standard approach for modeling discrete housing choices (McFadden, 1978). Bayer et al. (2007) showed how this framework can be applied to neighborhood sorting with heterogeneous preferences. The overlap between MARL and housing markets, however, is almost nonexistent. Existing algorithmic collusion studies use simplified Bertrand models that leave out the features that make housing distinctive: lease contracts, moving costs, income differences across tenants, and capacity constraints. This paper fills that gap.

3 Model

This section walks through the simulation environment. The model has three parts: (1) a tenant decision model that drives housing demand, (2) a physics engine that simulates the market month by month, and (3) landlord agents that set prices using either reinforcement learning or heuristic rules. All code is open-source at github.com/rowanmorkner/MARL_H-S.

3.1 Tenant Population

The simulation has $T = 2{,}000$ tenants, each with different incomes and housing situations. Incomes are drawn from the U.S. Census Bureau's American Community Survey (ACS) 2022 five-year Public Use Microdata Sample (PUMS) for the Sacramento–Roseville–Arden–Arcade, CA metropolitan statistical area. I filter the draw to civilian labor-force participants aged 18–65, so that dependents and retirees (whose income doesn't reflect earnings capacity) don't end up in the tenant pool. This gives a right-skewed earnings distribution that looks roughly like what you see in U.S. metropolitan labor markets.

Each tenant i has four attributes that matter for pricing:

1. **Income** (y_i): Yearly income, drawn as described above. Fixed for the duration of an episode.
2. **Current rent** (r_i): The rent tenant i currently pays (0 if homeless).
3. **Landlord ID** (ℓ_i): Which landlord tenant i is renting from (-1 if homeless). Homeless tenants act as shoppers and can move back into housing whenever a landlord is affordable.
4. **Lease timer** (τ_i): Months remaining on the current lease. When $\tau_i \leq 0$, the tenant becomes a *shopper* and evaluates all $N+1$ options (the N landlords plus the outside option of exiting the market). Lease timers are staggered uniformly at initialization, so shopping is spread across the year instead of everyone shopping in the same month.

The tenant state also has two more attributes, brand preferences and commute cost, which I set to zero weight for all the experiments reported here. I did this to isolate the effect of the concentration and friction parameters I actually vary. Their role in the utility function is described in Section 9.

3.2 Housing Supply

The market has N landlords, each owning $U_j = \lceil T(1+\text{ESR})/N \rceil$ housing units, where ESR is the *excess supply ratio* (Section 9). ESR $= 0.07$ therefore means the housing stock is 7% larger than the tenant population, so in expectation 7% of units are vacant when every tenant is housed. Each landlord j has two key attributes:

- **Price** (p_j): Posted rent, set by the landlord agent each period.
- **Marginal cost** (c): A fixed per-unit monthly operating cost (utilities, maintenance, taxes), held constant within a configuration and the same across all landlords in that configuration.

3.3 Tenant Decision Model

Tenants choose where to live using a McFadden conditional logit discrete-choice model (McFadden, 1978), extended to capture a few features specific to rental housing. When a tenant's lease timer reaches zero ($\tau_i \leq 0$), the tenant becomes a shopper and evaluates $N + 1$ options: move to any of the N landlords, renew at the current landlord (if housed), or exit the market (the outside option). The lease utility is a function of the contract terms at the time of signing, so once a tenant signs a lease their utility under that landlord is fixed for the whole lease. Prices only re-enter the tenant's decision at renewal or move time.

3.3.1 Utility Specification

The utility of shopping tenant i choosing landlord j is:

$$U_{ij} = \underbrace{q_0}_{\text{quality}} - \underbrace{\beta \cdot \frac{p_j}{s_i}}_{\text{price disutility}} \underbrace{- \beta \cdot \frac{m}{L \cdot s_i} \cdot \mathbf{1}[j \neq \ell_i]}_{\text{switching cost}} - \underbrace{B(p_j, y_i)}_{\text{affordability burden}}$$

where:

- $q_0 = 5.0$ is baseline housing quality (a normalizer for the utility scale).
- β is the McFadden price coefficient (swept across $\{0.01, 0.02\}$ to vary demand elasticity).
- $s_i = y_i/12{,}000$ is income scale; dividing price by s_i makes the price disutility rise sharply as rent claims a larger share of income.
- m is the up-front moving cost, amortized over the lease term L so that the per-period moving penalty is m/L.
- $\mathbf{1}[j \neq \ell_i]$ indicates that tenant i would be switching landlords rather than renewing.

The **affordability burden** $B(p_j, y_i)$ is a soft penalty on rents that exceed a standard affordability threshold:

$$B(p_j, y_i) = \alpha \cdot \max\left(\frac{p_j}{y_i/12} - \theta_{\text{soft}},\ 0\right)^2$$

with soft threshold $\theta_{\text{soft}} = 0.30$ and steepness $\alpha = 50$. The 30% threshold is the standard U.S. Department of Housing and Urban Development cutoff for a "cost-burdened" renter household

(U.S. Department of Housing and Urban Development, Office of Policy Development and Research, 2017). Above 30%, the burden penalty ramps up quadratically, which is meant to be smoother than a hard cutoff. A hard ceiling at $\theta_{\text{hard}} = 0.50$ (HUD's "severely cost-burdened" threshold) sets utility to $-\infty$, meaning the unit is simply infeasible for that tenant.

3.3.2 Renewal Option

Tenants whose leases expire at their current landlord get a modified utility:

$$U^{\text{renew}}_{i,\ell_i} = q_0 - \beta \cdot \frac{p^{\text{renew}}_{\ell_i}}{s_i} + \kappa - B(p^{\text{renew}}_{\ell_i}, y_i)$$

The incumbency bonus $\kappa = 2.5$ captures status quo bias and the hassle of moving (Genesove & Mayer, 2001). Importantly, renewal incurs no switching cost.

3.3.3 Outside Option

The outside option (exiting the market) has utility:

$$U_{i,\text{out}} = u_{\text{out}} - \beta \cdot \frac{c_{\text{out}}}{s_i}$$

where $u_{\text{out}} = -6.0$ is the quality of homelessness and $c_{\text{out}} = \$800$ is its monthly cost. This gives the model a smooth downward-sloping demand curve: as rents rise, lower-income tenants gradually exit instead of all dropping out at once.

3.3.4 Choice Probabilities

Given the utility values, tenant i picks option j with probability:

$$\Pr(j \mid i) = \frac{\exp(U_{ij})}{\sum_{k=0}^{N} \exp(U_{ik})}$$

using the numerically stable softmax (subtracting the maximum utility before exponentiation). Choices are then sampled from this distribution each period.

3.3.5 Capacity Enforcement

After all shoppers make their choices, any landlord whose units are oversubscribed triggers a cascade: excess tenants (randomly selected) get reassigned to landlords with remaining capacity, or become homeless if no capacity exists anywhere.

3.4 Simulation Dynamics

The physics engine runs at monthly frequency. Each timestep goes through six stages:

1. **Lease tick**: All lease timers decrement by 1.
2. **Shopper identification**: Tenants with $\tau_i \leq 0$ enter the market.
3. **Utility computation**: Full $(S \times N + 1)$ utility matrix computed for S shoppers.
4. **Choice sampling**: Stochastic logit choice for each shopper.
5. **State update**: Tenant states (landlord ID, rent, lease timer) updated.
6. **Capacity enforcement**: Overflow cascade for oversubscribed landlords.

Everything is vectorized in NumPy with no Python loops in the hot paths (except the capacity overflow cascade), which keeps the simulation fast enough for large parameter sweeps.

3.5 Landlord Agents

3.5.1 PPO Agents (Treatment)

Each landlord is an independent agent trained with Proximal Policy Optimization (PPO) (Schulman et al., 2017). The key design choices are as follows:

Observation space ($\mathbb{R}^5$): Each agent observes five normalized features:

1. Own occupancy rate $\in [0, 1]$
2. Own rent (log-normalized): $\log(p_j)/\log(p_{\max})$
3. Lease expiry risk: fraction of tenants whose leases expire next period
4. Market mean rent (log-normalized)
5. Market vacancy rate

Action space (Discrete(5)): Five multiplicative price adjustments applied to the current price:

Action	Multiplier	Interpretation
0	0.95	Panic cut (−5%)
1	0.99	Small cut (−1%)
2	1.00	Hold
3	1.01	Small hike (+1%)
4	1.05	Aggressive hike (+5%)

Reward (landlord objective function): At each month t, landlord j receives the normalized profit

$$r_{j,t} = \frac{\pi_{j,t}}{\hat{R}}, \qquad \pi_{j,t} = \sum_{i \in \mathcal{H}_{j,t}} \tilde{r}_{i,t} - c \cdot U_j,$$

where $\mathcal{H}_{j,t}$ is the set of tenants currently housed with landlord j, $\tilde{r}_{i,t}$ is tenant i's contracted rent (the posted rent at the time they signed their lease, which may differ from the landlord's current posted rent $p_{j,t}$), U_j is the landlord's unit count (the fixed cost is charged on all units regardless of occupancy), c is the common marginal cost, and $\hat{R}$ is a per-configuration normalizer equal to expected monthly revenue at the initial rent minus expected monthly cost ($\hat{R} = \max(p_0 \cdot T/N - c \cdot U_j,\ 1)$). Because rents are locked in at lease signing, revenue in any given month reflects a mix of recently-signed and older contracts rather than just the current posted rent times occupied units. The normalization keeps the effective reward scale roughly the same across configurations, which helps PPO training stay stable when rent levels differ across sweeps. The agent's actual objective is discounted cumulative normalized profit, $\sum_t \gamma^t r_{j,t}$, with discount $\gamma = 0.99$. This is the same profit-maximizing objective that standard industrial-organization models of pricing use.

Training: A single shared-parameter PPO policy is trained across all N agents at once using SuperSuit's `pettingzoo_env_to_vec_env_v1`, which presents each agent as an independent sub-environment. In other words, one policy network learns from all landlords' experience together, and then each landlord queries that same network using its own local observation to get its action. This is the Independent PPO (IPPO) setup from Yu et al. (2022). It directly models a common algorithmic pricing service: one vendor trains a single algorithm on data from competing properties, and every subscriber gets rent recommendations based on their own occupancy, current rent, and local market conditions. Landlords don't talk to each other, just like how subscribers to a real pricing platform get independent recommendations from the same underlying model.

PPO hyperparameters: learning rate 3×10^{-4}, batch size 1,920, discount factor $\gamma = 0.99$, network architecture $(64, 64)$ fully connected with 16 parallel environments. Training runs for 1,500 episodes $\times$ 240 timesteps per episode = 360,000 steps per configuration.

Domain randomization: Initial rents are randomized uniformly within ±20% of a log-uniformly sampled base rent (itself drawn within ±20% of $800), which helps the policy generalize rather than overfitting to one particular starting configuration.

3.5.2 Competitive Baseline (Occupancy Rule)

The competitive baseline is a simple rule-based agent that adjusts prices based on how its occupancy compares to a 7% vacancy target. I picked 7% because it's roughly the national rental vacancy rate in the Census Bureau's Housing Vacancy Survey for 2020–2023 (average ≈ 6.8%), and because it matches the largest value in the ESR sweep, so the rule-based agent is targeting the same kind of vacancy the environment can actually produce. The decision rule is:

$$a_j = \begin{cases} \text{Panic cut} & \text{if occupancy} < \text{target} - 0.05 \\ \text{Small cut} & \text{if occupancy} < \text{target} \\ \text{Hold} & \text{if occupancy} = \text{target} \\ \text{Small hike} & \text{if occupancy} > \text{target} \\ \text{Aggressive hike} & \text{if occupancy} > \text{target} + 0.05 \end{cases}$$

The logic is straightforward: cut prices when vacancies are high, raise them when demand is strong. This agent serves as the benchmark against which I evaluate PPO pricing behavior.

4 Experimental Design

4.1 Parameter Sweep

To systematically investigate how market structure affects pricing equilibria, I conduct a full-factorial parameter sweep across five dimensions:

Parameter	Values	Rationale
Number of landlords (N)	2, 5, 10, 20	Market concentration
Marginal cost (c)	\$200, \$350, \$500	Profit margin space
Moving cost (m)	\$750, \$1,500	Switching friction
Excess supply ratio (ESR)	0.02, 0.04, 0.07	Supply tightness
Price sensitivity (β)	0.01, 0.02	Demand elasticity

The full factorial yields $4 \times 3 \times 2 \times 3 \times 2 = 144$ unique configurations. Lease terms are fixed at 6 months, a common short-term-lease length in U.S. metro rental markets. I picked the parameter values to span a realistic range without blowing up the sweep size. $N \in \{2, 5, 10, 20\}$ covers duopoly through competitive, roughly matching the range of landlord concentrations actually observed in U.S. metro submarkets. Marginal costs of \$200–\$500/month span typical per-unit operating costs (utilities, maintenance, taxes) for multifamily rentals. Moving costs of \$750–\$1,500 span roughly one to two months of rent at the initial rent level of \$800. The ESR range covers tight-to-normal national rental vacancy rates (roughly 2% to 7%, per the Census Housing Vacancy Survey). Price sensitivity values of 0.01 and 0.02 span the range at which the McFadden logit gives economically reasonable own-price elasticities at the initial rent level. All configurations use Sacramento, CA Census income distributions and the default parameters listed in Section 3.

4.2 Evaluation Protocol

For each trained PPO model, I run a single evaluation episode under two conditions with a matched random seed:

1. **Control (IPPO)**: The trained PPO policy controls all N landlords, acting deterministically.
2. **Baseline**: N independent Occupancy-Rule agents control the landlords.

Using the same seed means both conditions see identical tenant populations, income draws, and initial conditions, so the two runs can be compared directly. Each evaluation episode runs for 240 simulated months (20 years). I use the steady-state period ($t \geq 120$, i.e., months 121–240) for the cross-sectional analysis, which gives the first 10 years for transient dynamics to wash out before I start collecting statistics.

5 Results

I report steady-state outcomes from the 144-configuration sweep. Cross-sectional statistics and regressions use landlord-level means over the final 120 simulated months (the steady-state window, $t \geq 120$). Time-series analysis uses the full evaluation trajectory. All tables and figures referenced below are collected in the Figures and Tables section at the end of the paper.

5.1 Summary Statistics

Table 3 reports steady-state summary statistics by pricing condition and number of landlords. Across all market concentrations PPO agents post substantially higher mean rents than the occupancy-rule baseline, and the Lerner index (the share of price above marginal cost) is materially higher under PPO in every row. At $N = 2$, mean PPO prices saturate at the simulator's administrative price cap (\$10,000; see Section 9) in every configuration. This is a boundary outcome imposed by the simulator rather than a market-clearing equilibrium: the duopoly PPO agents have learned that any price below the cap is accepted, and the action space prevents them from going higher. Duopoly results should therefore be read as a lower bound on what the algorithm would do with an unbounded action space, not as a clean equilibrium price. For $N \geq 5$, prices sit below the cap and the regime is no longer boundary-constrained. The premium declines consistently and rapidly with the number of competitors: by $N = 20$, PPO mean prices are within a few percent of the competitive baseline. Occupancy rates fall monotonically with the PPO markup, as expected when prices rise against a demand curve that is downward-sloping over the outside-option margin.

5.2 Price Premium by Market Structure

Figure 1 plots mean steady-state prices by number of landlords for both conditions. The central finding is visible at a glance: PPO prices decrease sharply with N, consistent with the standard oligopoly prediction that algorithmic pricing power is greatest in concentrated markets. The competitive baseline shows a much weaker (and slightly opposite-signed) relationship with N, as the occupancy-targeting heuristic is insensitive to market structure by construction. The ratio of PPO to baseline mean prices falls from roughly $14\times$ at $N = 2$ to near $1\times$ at $N = 20$, meaning the PPO markup is almost entirely eliminated once the market becomes sufficiently fragmented.

5.3 Price Distributions

Figure 2 shows the full kernel density of steady-state prices by condition and N. Baseline prices cluster tightly near the competitive equilibrium for every N. PPO distributions are right-skewed and shift leftward as N increases, consistent with the mean patterns above but also revealing substantial within-configuration variance: the sustained supra-competitive price level is not a razor-thin attractor but a broad high-price region.

5.4 Convergence Dynamics

Figure 3 shows mean PPO market price over time during the evaluation period, stratified by N. Prices converge within approximately 60–80 months of evaluation in every condition, with more concentrated markets converging to higher steady-state levels. The convergence is monotone from the randomized initial rents upward in the concentrated markets and roughly flat in the competitive ($N = 20$) case. The 120-month steady-state window used for the cross-sectional analysis is well after convergence.

5.5 Regression Analysis

5.5.1 Main Specification

I estimate the following panel regression on the landlord-level steady-state means:

$$\log(p_{ij}) = \beta_N \log(N_j) + \beta_c \log(c_j) + \beta_m \log(m_j) + \beta_E \log(\mathrm{ESR}_j) + \beta_{\mathrm{ps}} \log(\beta_{\mathrm{ps},j}) + \alpha_i + \varepsilon_{ij}$$

where p_{ij} is the mean steady-state price set by landlord i in configuration j, N_j is the number of landlords, and the remaining regressors are the swept market-structure parameters. α_i are landlord fixed effects (included in the within-IPPO specifications), and standard errors are clustered at the experiment level. Unobserved determinants of landlord-specific pricing behavior (for example, weight-initialization noise in the shared policy or draws of the tenant population that make certain landlords more attractive at random) are absorbed by α_i and the residual ε_{ij}.

Table 4 reports two specifications. Column (1) pools both conditions and identifies the PPO effect via the `PPO Agent` indicator and its interaction with $\log(N)$. Column (2) restricts to the IPPO sample and adds landlord fixed effects, which absorb time-invariant landlord heterogeneity; the `PPO Agent` indicator drops because it is constant within this subsample. The PPO-conditional elasticity of price with respect to N is large and negative in both specifications. In the pooled model (column 1) the baseline $\log(N)$ slope is small and slightly positive (+0.08), while the `PPO Agent` $\times \log(N)$ interaction is -1.25, so the implied PPO slope is -1.18. The within-IPPO specification gives $\hat{\beta}_N = -1.18$ (column 2), consistent with the pooled implied slope. Taken together, a doubling of N reduces the PPO price by roughly 55% ($1 - 2^{-1.18} \approx 0.56$). Marginal-cost pass-through into log price is modest: $\hat{\beta}_c$ is 0.08 (pooled) and 0.17 (IPPO FE), statistically indistinguishable from zero in the pooled specification. Pass-through therefore lies well below unity, consistent with PPO agents pricing largely on market structure rather than on costs. One caveat on the interpretation: at $N = 2$ every PPO price pins to the \$10,000 action-space cap regardless of marginal cost, so roughly a quarter of the sample has zero cost pass-through by construction; this mechanically deflates $\hat{\beta}_c$. Pass-through is therefore unlikely to be literally zero, but the weight of evidence is that it is meaningfully below unity rather than near-complete. The `PPO Agent` $\times \log(N)$ interaction in column (1) is large, negative, and precisely estimated: the concentration effect operates almost entirely through the PPO condition, as expected given the competitive baseline's near-flat relationship with N.

5.5.2 Friction Interactions

To test whether housing-market frictions moderate the concentration effect, I interact $\log(N)$ with each friction parameter in turn. Table 5 reports three IPPO fixed-effects specifications, one for each friction (moving cost, excess-supply ratio, and tenant price sensitivity), with the Wald F-statistic for the joint null that the interaction coefficient equals zero. Across all three frictions, the interaction coefficients are small in magnitude and statistically indistinguishable from zero. The Wald F-statistics are 0.39 (moving cost), 0.77 (excess-supply ratio), and 0.22 (price sensitivity), none significant at conventional levels, and the combined effect of $\log(N)$ evaluated at the sample mean of each friction is essentially identical to the base IPPO FE estimate (-1.18). The housing frictions modeled here do not meaningfully move the concentration-price slope. Figure 4 visualizes this by stratifying the N-slope by friction level; the slopes overlap tightly.

The full effect of $\log(N)$ is of course the sum of the main effect and the interaction effect evaluated at a specific friction level, not the interaction in isolation; Table 5 reports both. Even at the extremes of the friction grid, the implied concentration elasticity varies by less than 0.1 across friction levels within each specification.

5.6 Occupancy and Welfare Effects

Figure 5 shows mean occupancy rates by condition and N. Higher PPO prices translate into lower occupancy, with the gap between PPO and baseline occupancy narrowing as N rises and prices converge. The most extreme case is $N = 2$, where PPO agents post prices at the administrative cap and mean occupancy collapses to 9%. In this regime the algorithm has effectively learned to serve a small, high-income slice of the tenant distribution at very high rents rather than fill the available units. By $N = 20$ occupancy converges to 93%, essentially matching the baseline. Although the 50%-of-income hard affordability ceiling shapes the extreme tail of the demand response, in the configurations studied here the soft burden penalty does most of the work: occupancy declines smoothly with price rather than collapsing at a threshold.

Figure 6 shows mean monthly profit. PPO agents extract substantially higher profits than the baseline in concentrated markets despite lower occupancy: the price effect dominates the volume effect. The profit gap also narrows with N and approximately closes at $N = 20$, consistent with the pricing convergence in Figure 1.

6 Discussion

The results above show a pretty consistent picture: a shared PPO pricing policy learns inflated prices when deployed across competing landlords in this housing-market simulation, and the size of the premium falls with the number of competitors in a way that barely changes when I vary the housing-specific frictions. This section walks through the economic intuition behind the results, connects them to existing theory and evidence, and lays out the limitations.

6.1 Implications for Algorithmic Pricing in Housing

The finding that a shared pricing algorithm learns rents above competitive levels is directly relevant to the current policy debate around algorithmic pricing in rental markets. The shared-parameter IPPO setup models the hub-and-spoke structure at the center of the DOJ's case against RealPage (U.S. Department of Justice, 2024): a single algorithm trained on pooled market data, issuing pricing recommendations to each competing landlord based on local conditions. In my results, this pricing power is driven almost entirely by how many competing landlords there are. The number of landlords is the dominant predictor of the price premium, and the housing frictions I tested don't meaningfully change that relationship.

The null result on frictions is itself interesting. Some frictions should make collusion easier (high moving costs trap tenants with an incumbent landlord) and others should make it harder (tight supply limits how far prices can rise before running out of renters). It looks like these two effects roughly cancel out in my setup. The interaction coefficients are small compared to the main $\log(N)$ effect and the Wald tests don't reject zero (see Table 5). Moving costs, supply tightness, and demand elasticity shift the level of prices a bit, but they don't change the slope of the concentration-price relationship in any meaningful way.

These results also connect to the tacit-vs.-explicit-collusion distinction (see Section 9). Nothing in the simulation lets landlords send messages, watch each other's policies, or agree to a pricing rule. The algorithm gets to supra-competitive prices purely through the repeated-game dynamics over pooled data and a shared objective. That lines up with the tacit-collusion concern raised by Harrington (2018) and Ezrachi & Stucke (2016): the harm to consumers ends up looking a lot like harm from explicit coordination, even though no provable agreement exists.

The clearest policy implication is that market concentration is the main lever. Friction-specific interventions (rent control that indirectly changes moving costs, or zoning reforms that change supply elasticity) might move the level of prices a bit, but they shouldn't be expected to break the concentration channel that drives algorithmic markups in my results. The tools that do act on that channel are the familiar ones: merger scrutiny, entry promotion, and direct regulation of pricing platforms like what the DOJ is already doing in the RealPage case (U.S. Department of Justice, 2024).

6.2 Relationship to Existing Literature

My results mainly work as a robustness check on the existing algorithmic-collusion literature. Calvano et al. (2020) showed that independent Q-learning agents converge to supra-competitive prices in a simple Bertrand duopoly. This paper asks whether the same thing happens in a much richer environment: lease contracts that lock tenants in for multiple months, capacity constraints that bind when demand is high, tenants with different incomes drawn from ACS microdata, a soft affordability ceiling with quadratic penalties, and an explicit outside option to exit the market. The answer is yes: a shared PPO policy still learns inflated prices that decline with competition, basically the same pattern as in the frictionless setting.

I should be clear about what this paper doesn't do. The finding that algorithms raise prices in concentrated markets isn't new. Calvano *et al.* already showed this with Q-learning, and several other papers have replicated it with different algorithms and demand models (Calvano et al., 2020; Hettich, 2021; Klein, 2021; Schlechtinger et al., 2024). What I add is different. First, the environment: a housing market with leases, moving costs, capacity constraints, and ACS-calibrated tenant incomes, instead of a stylized Bertrand setup. Second, the interaction test: showing that the concentration effect holds basically the same way once you add all these housing-specific frictions.

The housing frictions I tested are the kinds of features that could plausibly have disrupted algorithmic coordination. Moving costs could segment the market and reduce the return to price-cutting. Lease contracts could slow the competitive response. Capacity constraints could create bottlenecks that distort pricing incentives. But none of them meaningfully change the competition-price slope that the algorithm ends up learning.

This paper also extends the algorithmic-collusion literature in two smaller methodological ways. Using PPO instead of Q-learning shows that supra-competitive prices aren't just an artifact of tabular methods — they persist with modern deep RL too. And the shared-parameter setup lines up with Ezrachi & Stucke (2016)'s Hub-and-Spoke framework, which is a more realistic model of a common pricing algorithm being deployed across competing firms than fully independent learners.

The way the price premium falls as N increases is consistent with Klein (2021)'s finding that collusion is harder to sustain with more competitors, and with the general oligopoly intuition going back at least to Tirole (1988). Even at $N = 20$, algorithmic prices are still marginally above competitive levels, but the premium is much smaller than in the duopoly case.

6.3 Limitations

This study has several important limitations.

Simulation vs. reality. The simulation is still a big simplification. Real housing markets have variable unit quality, new construction, rent control and zoning, information asymmetries, macroeconomic shocks, and people moving in and out of cities. How well these simulation results transfer to any specific real market is almost certainly limited. I am not claiming the results directly describe Sacramento. The claim is narrower: if the basic concentration-price result from simple oligopoly models holds up when you add realistic housing frictions, that's a robustness check on the existing literature, not an external validity claim about a specific metro.

Shared PPO as a pricing-service model. The shared-parameter IPPO setup captures the hub-and-spoke structure of platforms like RealPage (one algorithm, many subscribers, local recommendations), but there are real differences. PPO here learns from scratch by trial-and-error in the simulation, while real platforms like RealPage use their own historical transaction data and hand-built heuristics on top of any ML. The simulation shows that the shared-algorithm structure *can* produce inflated prices, but it is not direct evidence for the size of any specific platform's price effect.

Competitive baseline choice. The occupancy-targeting rule is a heuristic, not an analytical Nash equilibrium. It represents a plausible competitive strategy (price down when vacancies are high, up when demand is strong) and its 7% vacancy target is calibrated to the U.S. national rental vacancy rate, but the true competitive benchmark for this environment is unknown. The price premium should be interpreted relative to this specific baseline.

Single training seed. Each configuration is trained with one random seed, so the results don't capture variance from weight initialization or the early learning trajectory. MARL is non-stationary: each landlord is learning against opponents who are also learning, and tiny differences in initialization can push the agents to different equilibria. The clustered standard errors in my regressions soak up some of this noise, but they don't identify it on its own. Multi-seed replication would be cleaner, and I plan to do this for the final version.

Full adoption. Every landlord in the simulation subscribes to the same pricing service. Real markets have partial adoption: some landlords use competing algorithms, some use hand-tuned rules, some make pricing decisions by hand. The full-adoption case is an upper bound on what a common algorithm can do to prices. Partial adoption would almost certainly produce smaller premiums.

6.4 Future Directions

A few extensions would make these findings stronger:

1. **Deviation experiments** (Calvano-style). Force one agent to temporarily undercut, then see whether the others retaliate. This would help tell the difference between prices that are actively defended (tacit collusion) and prices that are just locally optimal given what everyone else is doing.

2. **Partial adoption.** Mix PPO-subscribing landlords with rule-based or independently trained landlords. This would show how the pricing premium depends on the market share of the common algorithm, and whether the shared-algorithm markup survives when some of the competitors aren't using the service.

3. **Shared vs. independent comparison.** I'm currently running a comparison between the shared-parameter IPPO setup in this paper and a fully-independent version where each landlord trains its own policy. Preliminary results on a subset of the sweep suggest the shared architecture doesn't actually amplify collusion relative to independent learners, but I'm saving the full comparison for the final version.

4. **Cross-training.** Train agents in one kind of market environment and then compete them in a different one. This tests whether learned strategies are robust to opponents they haven't seen and to shifts in market conditions.

5. **Dynamic supply.** Introduce endogenous construction decisions. This tests whether high PPO prices attract entry that eats away at the premium over time, which is the classic long-run competitive adjustment that my fixed-stock model can't capture.

7 Conclusion

This paper investigates how market structure parameters shape pricing outcomes when competing landlords subscribe to a common algorithmic pricing service in a simulated housing market. Using a multi-agent simulation with Census-calibrated tenant populations, McFadden logit demand, and realistic housing frictions, I find that a shared PPO pricing policy consistently learns rents far above a competitive heuristic baseline. The price premium declines steadily as the number of competing landlords increases.

Three findings stand out. First, the number of competitors is the dominant driver of algorithmic pricing power. Duopoly PPO prices are more than an order of magnitude above twenty-firm PPO prices, and $\log(N)$ explains the bulk of the variation in the pricing regressions (the PPO-conditional elasticity is roughly -1.18, so a doubling of N cuts prices by about 55%). Second, housing-specific frictions, including moving costs, supply tightness, and demand elasticity, do not meaningfully reshape this concentration-pricing relationship. The interaction terms between $\log(N)$ and each friction parameter are small and statistically insignificant. The pattern predicted by simple Bertrand oligopoly models holds up even when the environment includes realistic housing market complexity. This acts as a robustness check on prior algorithmic collusion work (Calvano et al., 2020; Johnson et al., 2023): the core finding that shared algorithms raise prices in concentrated markets does not depend on the simplifying assumptions of those stylized settings, though I am not claiming direct transferability to any specific real metro. Third, and more tentatively, marginal-cost pass-through into prices is modest (estimated elasticities of 0.08–0.22) and mostly statistically insignificant, well below the near-unity pass-through one would expect from a textbook profit-maximizer. Part of this attenuation is mechanical: at $N = 2$, every PPO price pins to the simulator's action-space cap regardless of marginal cost, so roughly a quarter of the sample carries zero pass-through by construction. Taken with that caveat, the evidence points toward PPO agents pricing predominantly off market structure rather than off operating costs, though disentangling the behavioral story from the boundary artifact would require running the sweep with a higher (or no) price cap.

For policymakers, these results point to a straightforward implication: market concentration is the key risk factor for algorithmic pricing distortions in rental housing. Frictions like moving costs and tight supply do not create additional collusive risk beyond what the number of competitors already predicts. This suggests that policy attention is best directed at market structure itself, for example through merger scrutiny and entry promotion, rather than friction-specific interventions. Direct regulation of pricing platforms (U.S. Department of Justice, 2024) remains important, but the structural conditions that enable algorithmic price elevation are the same ones that have long concerned antitrust authorities in any oligopoly setting.

8 Figures and Tables

All figures and tables referenced in the text are collected in this section in the order in which they are cited. Each appears on its own page. Data are drawn from the steady-state window ($t \geq 120$ months, i.e., after the initial ten simulated years of transient dynamics) of the full-factorial 144-configuration sweep described in Section 4.

Summary Statistics

Table 3: Steady-state summary statistics by pricing condition and number of landlords ($t \geq 120$). PPO Agents refers to the trained shared-parameter IPPO policy (`condition = control`); Competitive Baseline refers to the occupancy-rule agent (`condition = baseline`). Observations are landlord-month panel cells.

Condition	N	Mean Price	SD Price	Occupancy	Vacancy	Lerner Index	Profit	Observations
Competitive Baseline	2	685	136	0.919	0.041	0.468	291384	8,640
Competitive Baseline	5	700	158	0.922	0.038	0.475	122323	21,600
Competitive Baseline	10	707	176	0.921	0.039	0.474	61804	43,200
Competitive Baseline	20	796	213	0.913	0.048	0.525	38321	86,400
PPO Agents	2	10000	0	0.087	0.909	0.965	542479	8,640
PPO Agents	5	5919	2576	0.345	0.638	0.929	488046	21,600
PPO Agents	10	1395	547	0.862	0.100	0.693	168879	43,200
PPO Agents	20	829	241	0.931	0.029	0.533	43103	86,400

Main Regression

Table 4: How market structure shapes steady-state PPO prices. Dependent variable is log(price). Column (1) pools PPO and baseline conditions with a PPO-by-log(N) interaction. Column (2) keeps only the PPO sample and adds landlord fixed effects. Standard errors clustered at the experiment level.

	log_price	
	Pooled + Interaction	IPPO (Landlord FE)
	(1)	(2)
Constant	3.410***	
	(0.6109)	
PPO Agent	3.731***	
	(0.1069)	
log(N)	0.0766***	-1.177***
	(0.0121)	(0.0394)
log(Marginal Cost)	0.0832	0.1663*
	(0.0517)	(0.0949)
log(Moving Cost)	0.0390	0.0896
	(0.0521)	(0.0967)
log(Excess Supply Ratio)	-0.4398***	-0.4109***
	(0.0356)	(0.0681)
log(Price Sensitivity)	-0.1883***	-0.3648***
	(0.0521)	(0.0967)
PPO Agent × log(N)	-1.253***	
	(0.0433)	
Observations	2,664	1,332
R^2	0.85788	0.83882
Within R^2		0.77438
Landlord fixed effects		✓

Friction Interactions

Table 5: Testing whether housing frictions change how concentration affects PPO prices. Each column interacts log(N) with one friction, on the IPPO sample with landlord fixed effects and experiment-level clustered standard errors. The Wald F-statistic (bottom rows) tests the joint null that the interaction coefficient is zero. Failing to reject means frictions don't meaningfully change the concentration-price slope. The 'Combined effect at mean' row gives $\widehat{\beta}_N + \widehat{\beta}_{\text{interaction}} \cdot \bar{x}$, i.e. the total log(N) effect evaluated at the sample mean of the interacting friction.

	log_price		
	Moving Cost	Supply Ratio	Price Sens.
	(1)	(2)	(3)
log(N)	-0.6622	-1.407***	-0.9420*
	(0.8261)	(0.2743)	(0.5142)
log(Moving Cost)	0.2742	0.0896	0.0896
	(0.2858)	(0.0970)	(0.0964)
log(Marginal Cost)	0.1663*	0.1663*	0.1663*
	(0.0948)	(0.0943)	(0.0946)
log(Excess Supply Ratio)	-0.4109***	-0.2355	-0.4109***
	(0.0681)	(0.1915)	(0.0679)
log(Price Sensitivity)	-0.3648***	-0.3648***	-0.5028*
	(0.0967)	(0.0970)	(0.2872)
log(N) × log(Moving Cost)	-0.0740		
	(0.1179)		
log(N) × log(Excess Supply Ratio)		-0.0702	
		(0.0801)	
log(N) × log(Price Sensitivity)			0.0553
			(0.1182)
Wald F (interaction)	0.39	0.77	0.22
Wald p-value	0.531	0.381	0.640
Combined log(N) effect at mean	-1.177	-1.177	-1.177
Observations	1,332	1,332	1,332
R^2	0.83917	0.83952	0.83902
Within R^2	0.77488	0.77537	0.77466
Landlord fixed effects	✓	✓	✓

Price Premium by Market Structure

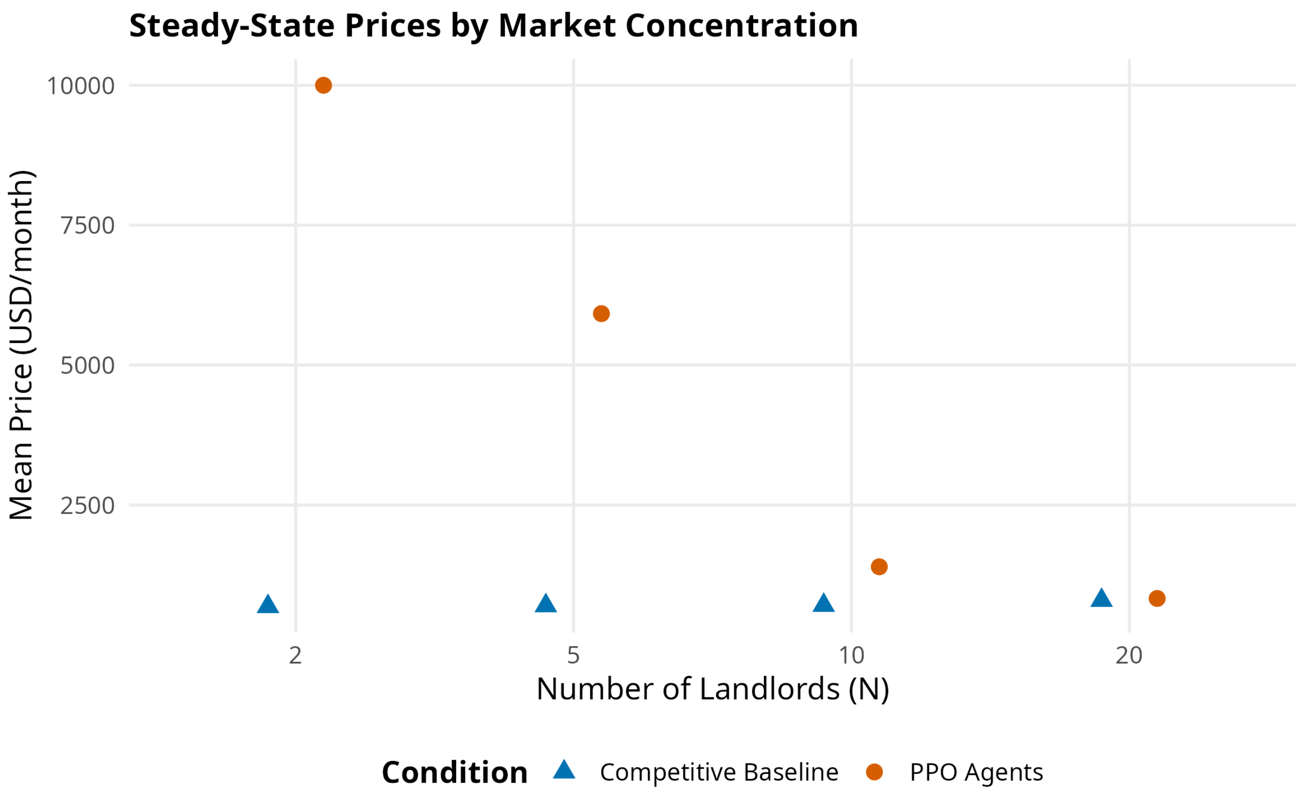

Figure 1: Mean steady-state price by number of landlords and pricing condition. PPO agents learn prices above the competitive baseline at most market concentrations, with the premium declining as competition increases.

Price Distributions

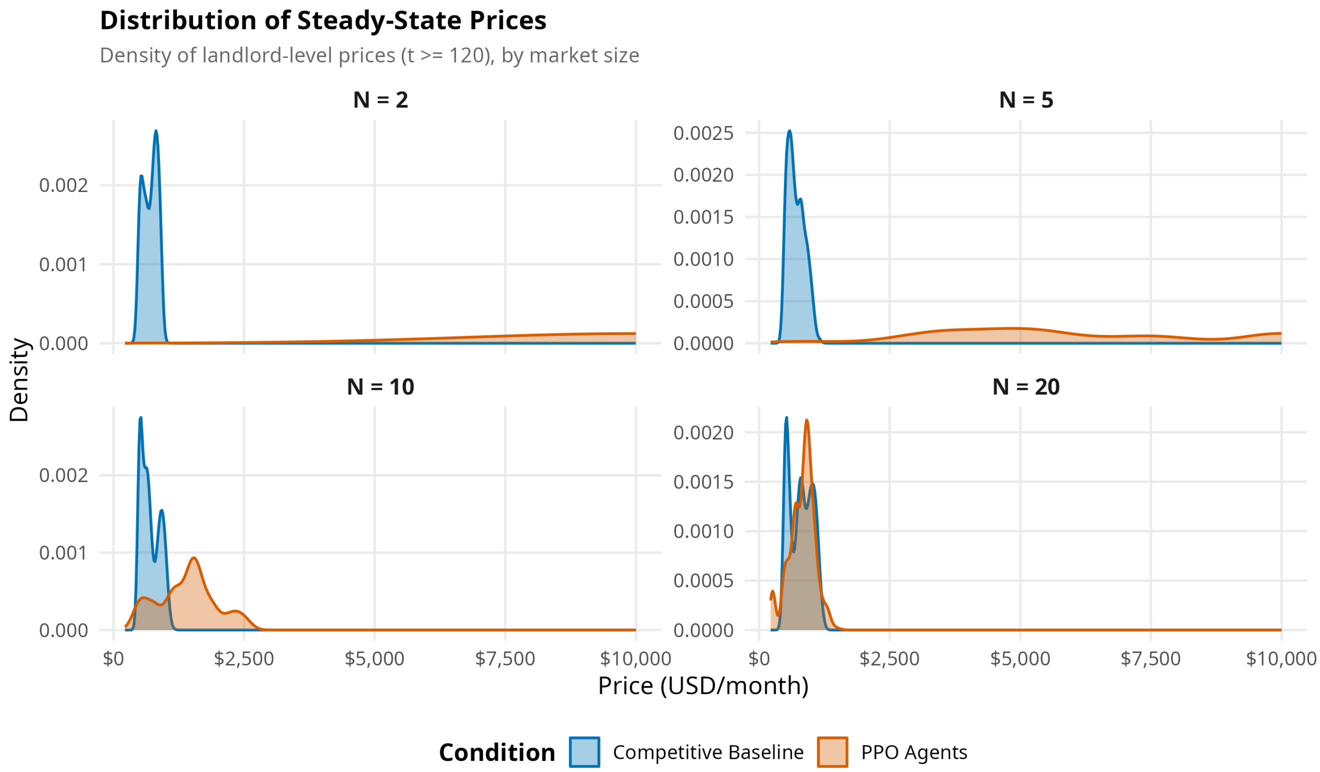

Figure 2: Density of steady-state prices by condition and number of landlords. PPO price distributions are right-skewed and shift leftward as N increases, while baseline prices cluster tightly near competitive levels.

Convergence Dynamics

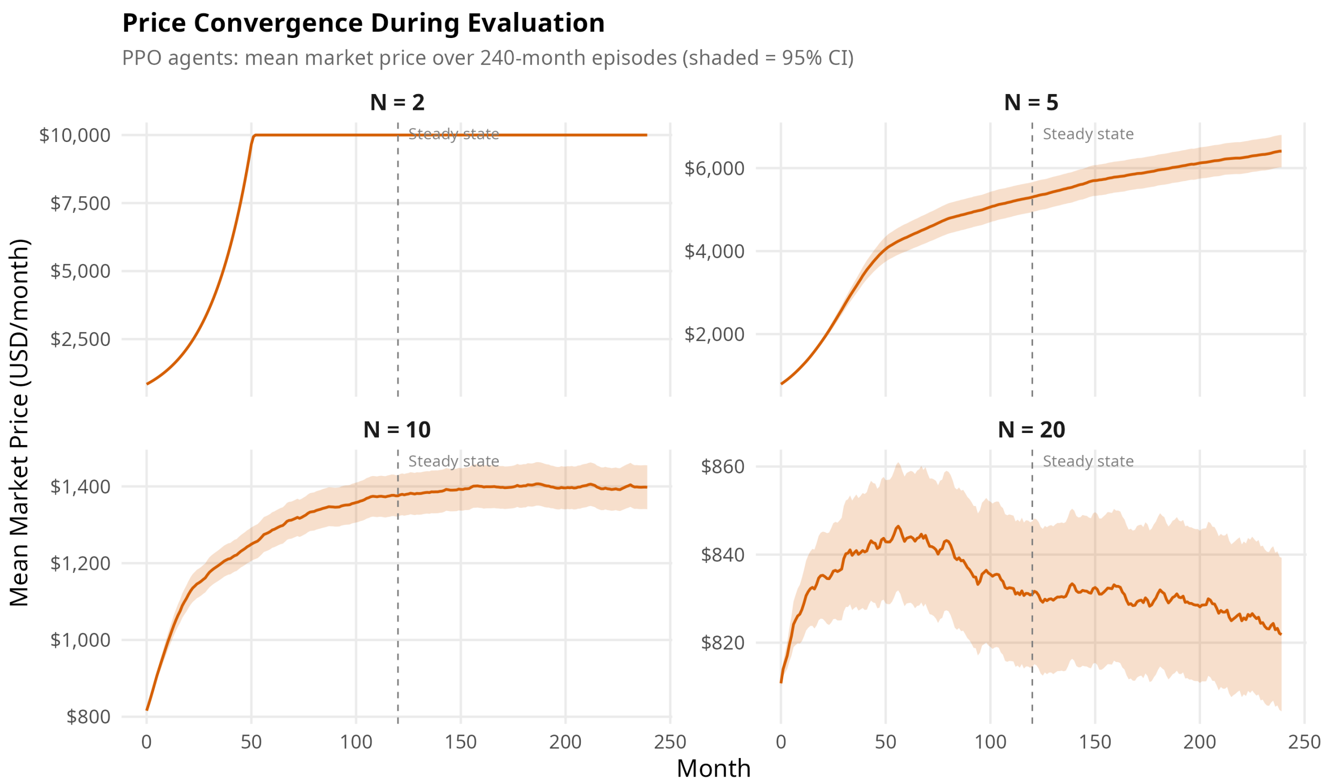

Figure 3: Mean market price over time for PPO agents, by number of landlords. Prices converge within approximately 60–80 months, with more concentrated markets converging to higher levels.

Friction Interaction Effects

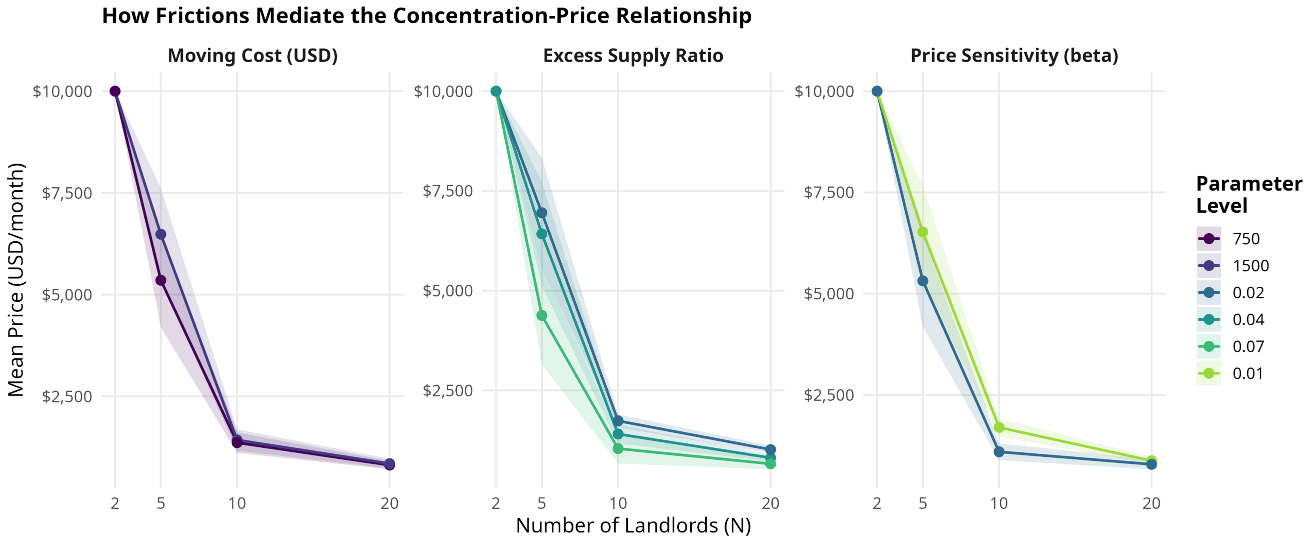

Figure 4: How housing-market frictions moderate the relationship between market concentration and PPO pricing. Each panel shows mean steady-state price by number of landlords, stratified by friction level. The slopes are nearly identical across friction levels within each panel, consistent with the statistically insignificant interaction coefficients in Table 3.

Occupancy by Market Structure

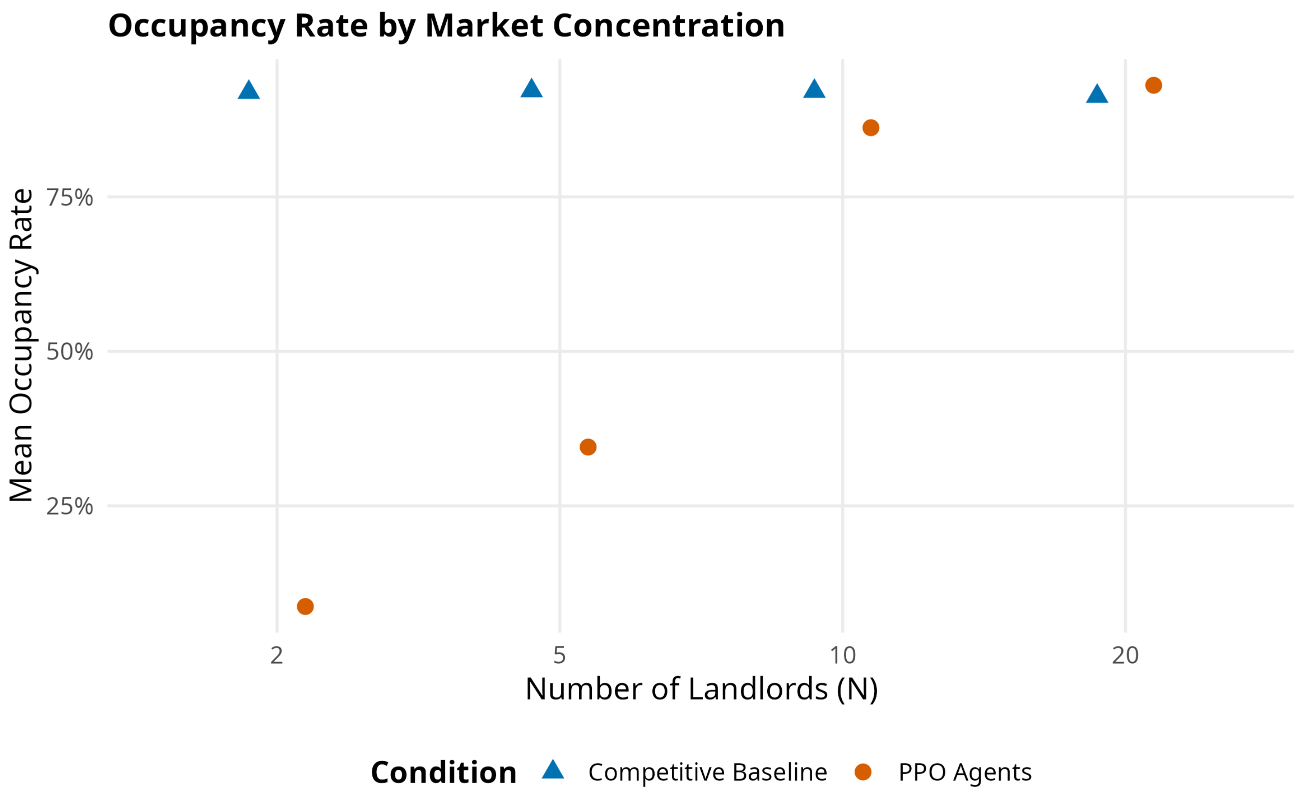

Figure 5: Mean occupancy rate by condition and number of landlords. Higher PPO prices are associated with lower occupancy, with the gap narrowing as competition increases.

Profit by Market Structure

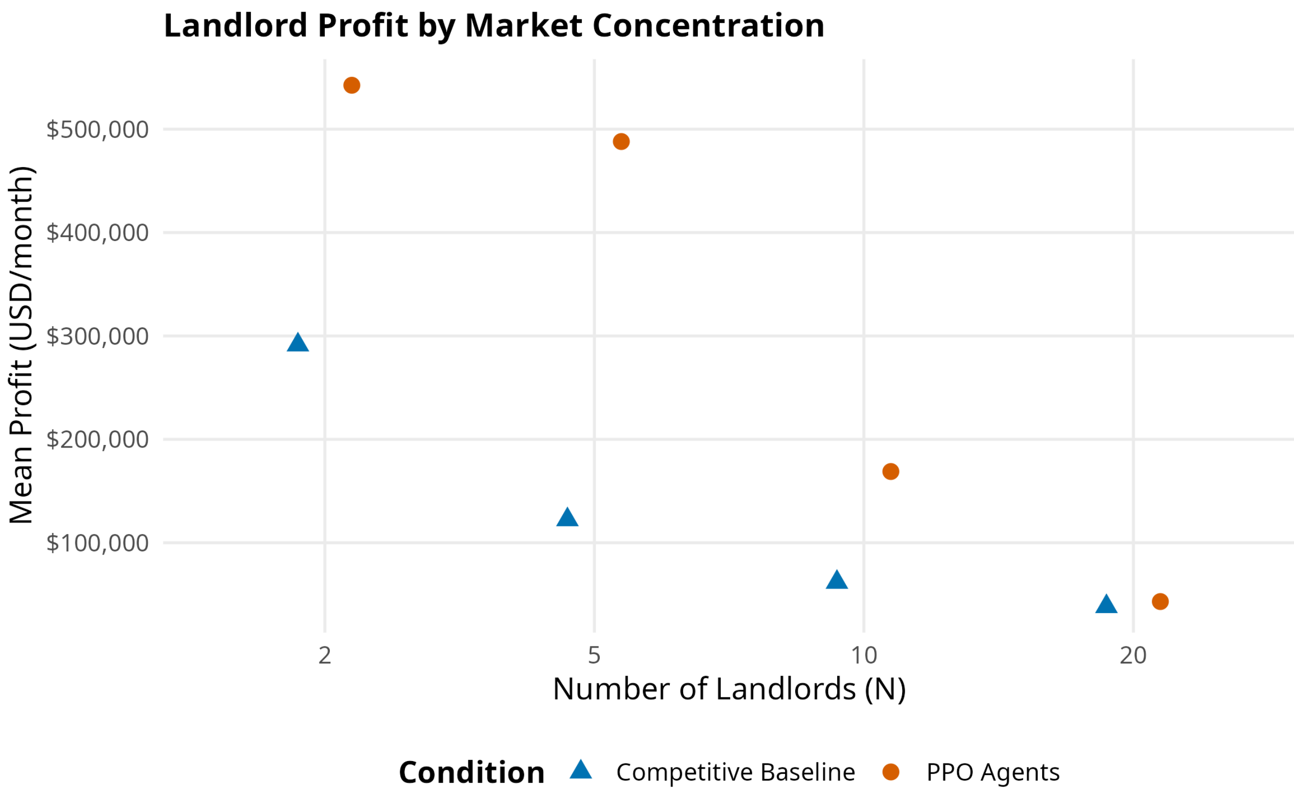

Figure 6: Mean monthly profit by condition and number of landlords. PPO agents extract substantially higher profits in concentrated markets despite lower occupancy.

9 Appendix A: Glossary of Non-Standard Terms

This appendix defines technical terms and a few words I use in a specific way in this paper. It is meant for readers who are coming in from one side (economics or ML) and don't have context on the other.

Administrative price cap / price floor. A numerical bound on the action space that prevents the simulator from evaluating degenerate prices ($0 or infinity). In this paper the cap is $10,000/month and the floor is $10/month. This is a *simulator-level* constraint, not the economic concept of a price ceiling (a binding regulatory cap below the market-clearing price). When the paper reports that duopoly prices "hit the ceiling," it refers to this administrative cap.

Excess supply ratio (ESR). The fraction by which the number of housing units exceeds the number of tenants in the simulation. $U_{\text{total}} = T \cdot (1 + \text{ESR})$. $\text{ESR} = 0.07$ therefore means that the housing stock is 7% larger than the tenant population, producing a 7% vacancy rate when every tenant is housed.

External validity. How much a result from a specific study setting carries over to the real world. Here, the question is whether findings from my 2,000-tenant simulation say anything useful about real rental markets like Sacramento.

Friction (housing-market). An institutional or structural feature that prevents instantaneous, costless reallocation of tenants across landlords. The frictions studied here are moving costs, lease terms, capacity constraints, and demand elasticity.

Hub-and-spoke (in algorithmic pricing). An industry structure in which a single vendor (the hub) trains a pricing algorithm and deploys it to multiple competing firms (the spokes), each of which submits local data and receives tailored recommendations. The central allegation in U.S. Department of Justice (2024) is that this structure enables coordination without direct communication between firms.

IPPO (Independent PPO). A multi-agent reinforcement learning paradigm in which each agent is treated as its own learner with its own policy updates. In the shared-parameter variant used here, all agents share a single neural-network policy trained on pooled experience. Each agent's actions still depend only on its local observations. The naming convention follows Yu et al. (2022).

Lerner index. $L = (p - c)/p$: the share of price that sits above marginal cost. Zero under perfect competition, positive whenever a firm has some market power. I use it as a simple way to compare markups across PPO and the baseline.

MARL (Multi-Agent Reinforcement Learning). The machine-learning subfield concerned with training multiple learning agents in a shared environment.

McFadden conditional logit. A discrete-choice model in which a decision-maker chooses among alternatives with probabilities proportional to $\exp(U_j)$, where U_j is the deterministic utility of alternative j. Standard workhorse model in empirical demand estimation (McFadden, 1978).

Parameter sweep. A procedure in which the simulation is run once per combination of values drawn from a grid of parameters. In this paper the grid has $4 \times 3 \times 2 \times 3 \times 2 = 144$ combinations.

PPO (Proximal Policy Optimization). A policy-gradient reinforcement-learning algorithm that improves the agent's policy while keeping each update close to the previous one (the "proximal" constraint) for training stability (Schulman et al., 2017).

Shared-parameter PPO. The specific variant of IPPO used here: a single neural-network policy shared across all landlord agents, updated on experience pooled from all of them. At deployment each agent queries the shared policy with its own local observation and receives its own action. This models an algorithmic pricing service that deploys the same underlying model to many competing subscribers.

Supra-competitive pricing. A price above the Nash-equilibrium competitive benchmark. The algorithmic-collusion literature uses this term to describe cases where learning agents end up holding prices above competitive levels without any explicit agreement between them.

Tacit vs. explicit collusion. *Tacit* collusion is when firms end up holding prices above competitive levels without ever agreeing to do so (usually through a repeated-game punishment structure). *Explicit* collusion is a direct, provable agreement to fix prices, like a cartel meeting. Algorithmic collusion sits somewhere in between: no humans are talking, but the shared algorithm and pooled data can still act as a coordinating mechanism, which is what raises the antitrust concern in Harrington (2018) and Ezrachi & Stucke (2016).

References

Assad, S., Clark, R., Ershov, D., & Xu, L. (2024). Algorithmic pricing and competition: Empirical evidence from the german retail gasoline market. *Journal of Political Economy*, *132*(3), 723–771.

Banchio, M., & Skrzypacz, A. (2022). Artificial intelligence and auction design. *Proceedings of the 23rd ACM Conference on Economics and Computation.*

Bayer, P., Ferreira, F., & McMillan, R. (2007). A unified framework for measuring preferences for schools and neighborhoods. *Journal of Political Economy*, *115*(4), 588–638.

Calder-Wang, S., & Kim, G. H. (2024). Coordinated vs. Efficient prices: The impact of algorithmic pricing on multifamily rental markets. *Working Paper.*

Calvano, E., Calzolari, G., Denicolò, V., & Pastorello, S. (2020). Artificial intelligence, algorithmic pricing, and collusion. *American Economic Review*, *110*(10), 3267–3297.

Ezrachi, A., & Stucke, M. E. (2016). *Virtual competition: The promise and perils of the algorithm-driven economy.* Harvard University Press.

Friedrich, P., Pasztor, B., & Ramponi, G. (2025). Learning collusion in episodic, inventoryconstrained markets. *Proceedings of AAMAS 2025*.

Genesove, D., & Mayer, C. (2001). Loss aversion and seller behavior: Evidence from the housing market. *The Quarterly Journal of Economics*, *116*(4), 1233–1260.

Harrington, J. E. (2018). Developing competition law for collusion by autonomous artificial agents. *Journal of Competition Law & Economics*, *14*(3), 331–363.

Hettich, M. (2021). Algorithmic collusion: Insights from deep learning. *CQE Working Paper 94/2021*.

Ivaldi, M., Jullien, B., Rey, P., Seabright, P., & Tirole, J. (2003). The economics of tacit collusion. *IDEI Working Paper*.

Johnson, J. P., Rhodes, A., & Wildenbeest, M. (2023). Platform design when sellers use pricing algorithms. *Econometrica*, *91*(5), 1841–1879.

Klein, T. (2021). Autonomous algorithmic collusion: Q-learning under sequential pricing. *The RAND Journal of Economics*, *52*(3), 538–558.

McFadden, D. (1978). Modeling the choice of residential location. In A. Karlqvist, L. Lundqvist, F. Snickars, & J. Weibull (Eds.), *Spatial interaction theory and planning models* (pp. 75–96). North-Holland.

Schlechtinger, M., Kosack, D., Krause, F., & Paulheim, H. (2024). By fair means or foul: Quantifying collusion in a market simulation with deep reinforcement learning. *Proceedings of IJCAI 2024*.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., & Klimov, O. (2017). Proximal policy optimization algorithms. *arXiv Preprint arXiv:1707.06347*.

Tirole, J. (1988). *The theory of industrial organization*. MIT Press.

U.S. Department of Housing and Urban Development, Office of Policy Development and Research. (2017). *Defining housing affordability*. HUD USER PD&R Edge Article. https://www.huduser.gov/portal/pdredge/pdr-edge-featd-article-081417.html

U.S. Department of Justice. (2024). *United states v. RealPage, inc.* Civil Antitrust Complaint, Case No. 1:24-cv-00710.

Vogell, H. (2022). Rent going up? One company's algorithm could be why. *ProPublica*.

White House Council of Economic Advisers. (2024). *The cost of anticompetitive pricing algorithms in rental housing*. CEA Issue Brief.

Yu, C., Velu, A., Vinitsky, E., Gao, J., Wang, Y., Bayen, A., & Wu, Y. (2022). The surprising effectiveness of PPO in cooperative, multi-agent games. *Advances in Neural Information Processing*

Systems, *35*.