Structural Econometrics and Reinforcement Learning

Pranjal Rawat, Georgetown University
John Rust, Georgetown University *

February 2025

Abstract

This paper explores the synergies between structural econometrics and reinforcement learning. Structural econometrics interprets observed economic choices as optimal decisions under constraints, enabling counterfactual prediction of behavior under rule changes. Reinforcement learning offers a framework for learning optimal policies in multi-step problems through exploration and exploitation. We identify opportunities for cross-fertilization between these fields. Structural econometrics can leverage reinforcement learning algorithms to solve previously intractable high-dimensional economic models and games. Inverse reinforcement learning provides econometricians with new methods to recover agents' objective functions from observed behavior. Reinforcement learning, in particular bandits, can be enhanced by incorporating economic theory and structural assumptions, accelerating learning and improving sample complexity by orders of magnitude. We review methodological connections, demonstrate applications across finance, industrial organization, public policy, and marketing. Both fields create new tools for inference and decision-making while tackling the shared challenges of the curse of dimensionality, equilibrium multiplicity, and identification.

Keywords: Structural Econometrics, Reinforcement Learning, Industrial Organization, Discrete Choice, Discrete Games

*Corresponding Author: Department of Economics, Georgetown University, jr1393@georgetown.edu.

1 Introduction

Economic decisions rarely occur in isolation. Rather, they unfold sequentially over time, with choices today influencing opportunities tomorrow. They also interact with the decisions of others. The temporal and strategic dimension of economic behavior makes it difficult to model mathematically. How can we understand and predict decisions in changing environments, evolving strategic competitors, and varying policy rules? Two methodological fields have grown to tackle this challenge: structural econometrics in economics and reinforcement learning in computer science. Although these fields have developed independently, they address similar problems through complementary approaches.

Structural econometrics, rooted in economic theory, interprets observed choices as optimal decisions made under institutional and technological constraints. By estimating the underlying preference and technology parameters driving these choices, econometricians can perform counterfactual analysis—predicting how agents would behave if the "rules of the game" changed. This ability to simulate policy counterfactuals makes structural models invaluable for economic policy design and evaluation. However, structural approaches have traditionally been limited by computational constraints in handling high-dimensional state spaces and strategic interactions.

Reinforcement learning (RL), meanwhile, has emerged as a powerful paradigm for sequential decision-making under uncertainty. RL generalizes dynamic programming by learning optimal policies directly from interaction with an environment, without requiring explicit specification of transition probabilities or reward functions. Recent advances in deep reinforcement learning have demonstrated remarkable successes from game play to robotics, language model alignment, and logistics optimization. These methods excel at finding approximately optimal policies in high-dimensional environments where traditional dynamic programming would be intractable.

Both fields evolved through parallel yet independent trajectories. Structural econometrics emerged from the Cowles Commission's work in the 1940s, establishing frameworks for connecting economic theory with statistical inference. It progressed through McFadden's discrete choice models in the 1970s and Rust's dynamic programming approaches in the 1980s to modern computational methods. Reinforcement learning traces its origins to early AI pioneers like Shannon and Minsky, crystallizing as a field through

Sutton and Barto's unifying work on temporal difference learning in the 1980s. Both disciplines experienced transformative computational advances in the 2010s—structural econometrics incorporating machine learning techniques for high-dimensional problems and reinforcement learning achieving breakthroughs with deep neural networks.

Methodologically, all dynamic programming and reinforcement learning algorithms can be subsumed in a single framework. Reinforcement learning algorithms can be understood as a generalization of dynamic programming that relaxes the requirement of knowing the transition probabilities and reward functions. This makes convergence slower, as model structure cannot be exploited, but allows RL to scale-up and overcome the curse of dimensionality that plagues DP. This helps us to understand why RL methods, under sufficient exploration of the state space, converge to the same optimal policies identified by traditional dynamic programming. We offer a few concrete examples to show how tabular and deep reinforcement learning can mitigate the curse of dimensionality in dynamic optimization problems and dynamic games.

We then examine how reinforcement learning has been applied to solve structural economic models. We look at experience-based equilibria in Asker et al. (2020), demonstrate how reinforcement learning principles can be used to define and compute equilibrium concepts in games with asymmetric information. In industrial organization, RL techniques have proven valuable for analyzing dynamic oligopoly models with endogenous innovation, entry/exit, and mergers—problems that were previously computationally prohibitive. We also explore applications in dynamic discrete choice modeling, where policy gradient methods combined with indirect inference provide new estimation approaches for models with continuous state variables and unobserved heterogeneity. In macroeconomics, deep reinforcement learning has enabled solution of general equilibrium models without relying on simplifying approximations.

The paper further investigates Inverse Reinforcement Learning (IRL)—a subfield that seeks to recover reward functions from observed behavior. We highlight the strong methodological parallels between IRL and structural estimation in economics. Both aim to infer underlying preferences from choice data, but approach the problem through different computational techniques. Maximum entropy IRL, in particular, shares deep connections with discrete choice models in econometrics, with the Boltzmann policy formulation corresponding precisely to multinomial logit specifications with Gumbel-distributed util-

ity shocks. Recent innovations like Sharma et al.'s (2018) adaptation of Hotz-Miller's conditional choice probability approach demonstrate how econometric methods can enhance IRL's computational efficiency, while IRL's flexible function approximation capabilities can benefit structural estimation of complex preferences.

Reinforcement learning in strategic settings forms another critical area of overlap. We examine how multi-agent reinforcement learning connects to evolutionary game theory through replicator dynamics, and review algorithms specifically designed for competitive environments. Methods like counterfactual regret minimization that have achieved superhuman performance in games offer powerful new approaches for modeling strategic interaction in economics, with applications from auction design to industrial organization and macroeconomic policy.

Economic theory provides valuable structure that can significantly enhance reinforcement learning algorithms. By incorporating domain-specific knowledge like monotonicity in demand curves, unimodality in auction revenues, and low-rank structures in multi-product markets, reinforcement learning can achieve exponentially improved learning efficiency. For dynamic pricing problems, economic structure enables algorithms to achieve logarithmic rather than square-root regret. In resource-constrained optimization settings, economic principles of marginal value versus marginal cost naturally map to dual formulations that balance exploration and exploitation. When modeling strategic interactions, economic equilibrium concepts provide stability guarantees and solution refinements. These structural assumptions not only accelerate learning but also produce more interpretable and generalizable policies, creating a bridge between theory-driven and data-driven approaches to decision-making.

Finally, we look at real world applications of SE and RL where these methodological innovations have delivered practical impact. We examine how structural econometric models have shaped antitrust policy, market design, program evaluation, and monetary policy, while reinforcement learning has begun to be applied to financial trading, energy management, transportation systems, recommendation systems, and marketing optimization.

Throughout this exploration, several common themes emerge. Both fields grapple with similar fundamental challenges: the curse of dimensionality in large state spaces, equilibrium multiplicity in strategic settings, and identification of underlying preferences

from choice data. For economists, reinforcement learning offers powerful computational tools to solve economic models. For computer scientists, economic theory provides principled structural assumptions that can accelerate learning, reduce sample complexity, enable identification, and enhance interpretability.

Our survey and interpretative essay seek to contribute in a couple of ways. Firstly, it seeks to provide a comprehensive update of relevant papers at the intersection of SE and RL. Most surveys between economics and computer science have focused on supervised learning (Athey and Imbens 2019). This survey differs from Charpentier et al 2021 by focusing on the methodological connections. It builds on Iskhakov et al. 2021, but covers many new topics like inverse reinforcement learning, using RL to solve equilibria of games, and using of SE to improve RL.

The remainder of this paper proceeds as follows. Section 2 traces the historical development of both fields. Section 3 establishes the connection between dynamic programming and reinforcement learning and offers concrete examples. Section 4 examines applications of reinforcement learning to solving structural economic models. Section 5 explores inverse reinforcement learning and its connections to structural estimation. Section 6 investigates reinforcement learning in game-theoretic settings, replicator dynamics and scalability. Section 7 discusses how structural assumptions can enhance online reinforcement learning or bandits. Section 8 surveys real-world applications across various domains. We conclude by identifying promising directions for future research at the interdisciplinary frontier.

2 Historical Developments

2.1 A Brief History of Reinforcement Learning

The reinforcement learning (RL) journey began with early computational work on game engines. Shannon (1950) created chess evaluation functions, anticipating programs that could "learn from mistakes." Minsky (1954)'s thesis connected neural networks to Skinnerian conditioning, introducing a "second memory" for outcome prediction—foreshadowing value functions. Samuel (1959)'s groundbreaking checkers program implemented rote learning and parameter adjustment based on evaluation differences, presaging temporal difference learning. Concurrently, Bellman (1957) and Howard (1960) developed dynamic

programming for sequential decision-making, establishing the mathematical foundation of optimality through backward induction in fully-specified environments. These early contributions identified the core challenges that would drive RL research: credit assignment (determining which actions lead to rewards), exploration (trying new actions while exploiting successful ones), and function approximation (generalizing across state spaces).

Reinforcement learning crystallized as a coherent field through Sutton and Barto's unifying work, which addressed the critical problem of learning without complete environmental models. Their temporal difference (TD) learning method (Sutton, 1988) solved the credit assignment problem by bootstrapping—updating value estimates based on subsequent predictions rather than waiting for final outcomes—a middle ground between Monte Carlo approaches and dynamic programming. This innovation enabled agents to learn incrementally from partial experiences, a fundamental capability for online learning. Building on this foundation, they synthesized diverse approaches (Q-learning, SARSA, actor-critic methods) under the framework of generalized policy iteration (GPI), where policy evaluation and improvement operate symbiotically (Sutton and Barto, 1998).

A significant early framework was the actor-critic architecture, introduced by Barto et al. (1983), which combined a policy representation (actor) with a separate value function (critic). This approach allowed agents to handle both discrete and continuous action spaces while providing a natural implementation of policy gradient methods. Watkins and Dayan (1992)'s Q-learning, with its convergence guarantees, allowed an agent to learn optimal policies despite following a different behavioral policy. This "off-policy" learning allowed the agent to re-use past experiences. Meanwhile, policy gradient methods like REINFORCE (Williams, 1992a) directly optimized policy parameters, being particularly valuable for continuous or high-dimensional action spaces where value-based methods struggled.

The multi-armed bandit problem, a simplified setting where agents make one-shot decisions under uncertainty, provided a formal framework for analyzing the exploration-exploitation trade-off. Lai & Robbins (1985) established regret lower bounds which was integrated into robust algorithms like Upper Confidence Bound (UCB) by Auer et al. (2002) which had finite-time regret guarantees.

Function approximation represented the next frontier, as traditional tabular methods couldn't scale to problems with large state spaces. Tesauro (1994)'s TD-Gammon

marked a breakthrough: a backgammon-playing program that achieved near-human-expert performance through self-play, using TD learning with neural networks as function approximators. This success demonstrated that RL could scale to domains with proper generalization mechanisms. However, function approximation introduced new stability challenges—the "deadly triad" of bootstrapping, off-policy learning, and function approximation could cause divergence and instability in learning.

Before the deep learning era, the RL toolkit expanded in different directions. Sutton (1990)'s Dyna architecture unified learning and planning by integrating a learned model with direct reinforcement learning, establishing the model-based RL paradigm that would later influence systems like AlphaGo. Hierarchical frameworks tackled the complexity of long-horizon problems: Sutton et al. (1999b)'s Options framework formalized temporal abstraction, allowing agents to reason at multiple time scales. Brafman and Tennenholtz (2002)'s R-MAX algorithm provided theoretical guarantees for efficient exploration through the principle of "optimism in the face of uncertainty," automatically balancing exploration and exploitation. Policy optimization advanced through Kakade (2001)'s Natural Policy Gradient, which improved learning efficiency by accounting for the concavity of the loss landscape through Fisher information. Ng and Russell (2000) formalized Inverse Reinforcement Learning for inferring reward functions from demonstrations, enabling preference learning from expert behavior—a subfield with exactly the same goals as structural econometrics. Bertsekas and Tsitsiklis (1996) bridged RL with control theory through their "Neuro-Dynamic Programming" framework, providing rigorous theoretical analysis and applications to operations research problems.

The deep RL revolution overcame the problem of function approximation and the "deadly triad" that had limited previous scaling efforts. Mnih et al. (2015)'s Deep Q-Networks (DQN) achieved human-level performance on Atari games by combining Q-learning with convolutional neural networks, learning directly from raw pixels without hand-crafted features. Critical innovations that stabilized this integration included experience replay (breaking correlations between consecutive samples) and target networks (reducing update instability). These techniques addressed key problems in applying RL to high-dimensional sensory inputs, enabling end-to-end learning from raw observations to actions. The breakthrough opened a floodgate of innovations, with Schulman et al. (2015)'s Trust Region Policy Optimization and Schulman et al. (2017)'s Proximal Policy

Optimization providing similarly stable and reliable policy learning methods.

Game-playing achievements demonstrated how combining deep RL with planning could solve problems of unprecedented complexity. Silver et al. (2016)'s AlphaGo defeated world champion Lee Sedol by integrating supervised learning from human games with reinforcement learning from self-play, marking a milestone that many experts believed was decades away. More remarkably, AlphaGo Zero (Silver et al., 2017) achieved superhuman performance without human data, learning entirely through self-play. Igami (2020) noted the economic parallels, interpreting AlphaGo's architecture as combining a supervised learning policy network (similar to Hotz-Miller's conditional choice probability estimator) with a reinforcement learning value network (analogous to conditional choice simulation methods).

The application of reinforcement learning to language models represents one of its most significant recent developments. Reinforcement Learning from Human Feedback (RLHF) emerged as the dominant paradigm for aligning large language models with human preferences, addressing the challenge of optimizing for implicit human values rather than easily-specified reward functions. Key implementations include OpenAI's InstructGPT (Ouyang et al., 2022), which showed a significantly smaller RLHF-tuned model outperforming much larger base models on human preference metrics, and Anthropic's Constitutional AI (Bai et al., 2022), which developed RL from AI Feedback to reduce the need for human evaluations. These applications demonstrate how an RL framework enables AI systems to better align with human intentions.

2.2 A Brief History of Structural Econometrics

Structural econometrics emerged from the Cowles Commission's groundbreaking work in the 1940s, which established the foundation for relating economic theory to statistical analysis. Haavelmo (1944) introduced the probabilistic framework that distinguished between structural and reduced-form parameters, formalizing the identification problem in simultaneous equations. Koopmans developed order and rank conditions determining which parameters are identifiable given model specifications, while Marschak emphasized the importance of "autonomy" in structural equations—relationships that remain invariant under policy interventions. Hood and Koopmans (1953) consolidated these advances into a coherent set of tools, introducing estimation techniques like limited and

full-information maximum likelihood. This methodological fusion of economic theory with statistical inference became the defining characteristic of structural econometrics.

The field advanced significantly through McFadden (1974)'s discrete choice framework, which formalized the connection between utility maximization theory and probabilistic choice behavior. Heckman (1979)'s sample selection model addressed estimation bias from non-random sample selection, treating participation decisions as part of the economic model. These static frameworks laid groundwork for structural estimation in labor economics, industrial organization, and public finance. By the early 1980s, researchers began transitioning from static to dynamic sequential decision problems. Wolpin (1984) modeled fertility decisions over time, and Miller (1984) developed dynamic job matching models.

A milestone was Rust (1987)'s nested fixed-point (NFXP) algorithm, which formulated and estimated a fully specified Markov decision process for engine replacement decisions. For each trial parameter vector, Rust solved the Bellman equation via fixed-point iteration, then maximized the likelihood of observed choices. This methodological breakthrough showed that full-solution estimation of dynamic discrete choice models was computationally feasible, inspiring subsequent innovations to address the "curse of dimensionality." Berry et al. (1995) revolutionized demand estimation with their random coefficients logit model for differentiated products, inverting market share data to recover underlying consumer utilities while allowing flexible substitution patterns with endogenous prices. This "BLP" approach became the workhorse for static demand estimation in industrial organization, later enhanced through techniques integrating micro-data (Petrin, 2002) and methodological refinements addressing numerical performance (Dubé et al., 2012).

The computational burden of repeatedly solving dynamic games spurred methodological innovations in the 2000s. Hotz and Miller (1993) introduced the conditional choice probability (CCP) method, showing that observed choice probabilities could be inverted to recover value function differences without repeatedly solving the dynamic programming problem. Bajari et al. (2007) adapted this insight to dynamic games through a two-step approach—first estimating policy functions from data, then finding structural parameters that rationalize observed behavior. Aguirregabiria and Mira (2007) proposed nested pseudo-likelihood estimation, iteratively refining two-step estimates to reduce bias. Pe-

sendorfer and Schmidt-Dengler (2008) developed value function difference methods using equilibrium conditions directly.

Recent advances have further expanded structural econometrics' capabilities. Weintraub et al. (2008) introduced oblivious equilibrium concepts to handle high-dimensional state spaces with many agents. Ciliberto and Tamer (2009) developed moment inequality methods addressing equilibrium multiplicity through partial identification. Advances in production function estimation (Ackerberg et al., 2015) refined proxy methods for handling endogeneity. The 2010s witnessed integration with machine learning techniques, using neural networks as function approximators for value and policy functions. Fernández-Villaverde et al. (2024) demonstrated how deep learning can mitigate the curse of dimensionality in solving high-dimensional economic models. Maliar et al. (2021) used neural networks trained via stochastic gradient descent for solving dynamic programming problems, while Atashbar and Shi (2023) applied deep reinforcement learning to macroeconomic models.

3 Planning and Learning

In this section, we connect Dynamic Programming (DP) with Reinforcement Learning (RL) methods through a concrete navigation example. Both methods attempt to find optimal policies in Markov Decision Processes (MDPs). An MDP is defined as a tuple (S, A, P, R, γ), where S denotes the state space, A the action or choice space, $P(s'|s, a)$ the transition probability function, $R(s, a, s')$ the reward or utility function, and $\gamma \in [0, 1]$ the discount factor. Policies π are conditional probabilities functions that tell us the probability of taking actions in states.

Following Sutton and Barto's formulation, Generalized Policy Iteration (GPI) refers to the interaction between policy evaluation and policy improvement processes. Policy evaluation makes the value of being in states, i.e. the an estimate of the expected discounted returns, when following the current policy. Policy improvement finds the best policy given a valuation of the states. When both processes stabilize, the resulting policy and value function is optimal.

3.0.1 Dynamic Programming

First, we look at two DP techniques. The solution is stored in value and policy tables, initialized at 0. which sweep through, updating all states at each iteration (full-width). They both look ahead one step (depth=1), and use the previous guess of the value function to update (bootstrapping). The learning rate is 1, i.e. it is a full update. At each iteration, Value Iteration (VI) computes the optimal value function through the Bellman optimality equation:

$$\text{Value Iteration:} \quad \forall s, V(s) \leftarrow \max_a \sum_{s'} P(s'|s,a)[R(s,a,s') + \gamma V(s')] \tag{1}$$

The max operator delivers policy improvement, while the assignment delivers value evaluation. By requiring transition probabilities and potential rewards across all actions and future states, VI is inherently model-based. Theoretically, VI converges to the optimal value function at a linear rate, with error decaying geometrically with ratio γ Bertsekas (1996). For any finite MDP with $\gamma < 1$, VI guarantees convergence to V^* from any initial V_0 Sutton and Barto (2018).

Policy Iteration (PI) Alternates between policy evaluation and improvement:

$$\text{Policy Iteration:} \quad V^\pi(s) = \sum_{s'} P(s'|s,\pi(s))[R(s,\pi(s),s') + \gamma V^\pi(s')] \tag{2}$$

$$\pi'(s) = \underset{a}{argmax} \sum_{s'} P(s'|s,a)[R(s,a,s') + \gamma V^\pi(s')] \tag{3}$$

This method separates policy evaluation and policy improvement, and begins with a random policy, and recursively finds the value of states for it. PI converges to an optimal policy in a finite number of iterations, with each policy improvement yielding a strictly better policy until optimality is reached Bertsekas (1996).

3.0.2 Reinforcement Learning

We now turn to Reinforcement Learning (RL) methods, which do not require knowledge of P and R functions but only require access to transitions and rewards only at those particular actions and states (i.e. a simulator). The first two are temporal-difference methods that learn value tables. They begin from an initial state and explore their environment one state at a time (width=1) and look ahead one step (depth=1), using

their previous estimate of the value function (bootstrapping). The actions are not selected greedily, but ϵ-greedily which allows for randomness and exploration of the state space. The updates also happen incrementally with the learning rate $\alpha < 1$.

Q-learning is an off-policy algorithm that updates action-values using individual sampled experiences, following the update rule:

$$\text{Q-learning:} \quad Q(s,a) \leftarrow Q(s,a) + \alpha[R + \gamma \max_{a'} Q(s',a') - Q(s,a)] \tag{4}$$

At each step, the agent observes a tuple (s, a, s', r), chooses actions using an ε-greedy policy (selecting random actions with probability ε and the best-known actions with probability $1 - \varepsilon$), and updates its value estimates. Over time, ε typically decreases, shifting from exploration to exploitation. Under conditions of infinite exploration, diminishing learning rates, and greedy-in-the-limit behavior, Q-learning converges to the optimal action-value function Q^* with probability 1 (Watkins and Dayan, 1992).

SARSA for State-Action-Reward-State-Action updates action-values using the action selected by the current policy for the next state:

$$\text{SARSA:} \quad Q(s,a) \leftarrow Q(s,a) + \alpha[R + \gamma Q(s',a') - Q(s,a)] \tag{5}$$

Unlike Q-learning, SARSA computes the Bellman error based on actual behavior (next action chosen by the policy) and not greedy-behavior, making it on-policy i.e. the same policy that generated the samples is being updated. It similarly uses ε-greedy exploration to generate samples (s, a, s', r, a'). SARSA converges to the optimal Q^* with probability 1 when coupled with a policy that becomes greedy in the limit, provided every state-action pair is visited infinitely often Singh et al. (2000).

Monte Carlo methods are on-policy methods that learn from complete episodes, updating action-values based on observed returns:

$$\text{Monte Carlo:} \quad Q(s_t,a_t) \leftarrow Q(s_t,a_t) + \alpha[G_t - Q(s_t,a_t)] \tag{6}$$

They begin from random policies, which are traced out till termination (reaching the goal or being truncated), and then updates are made to each state-action visited. Here, the return $G_t = \sum_{k=0}^{T-t-1} \gamma^k R_{t+k+1}$ represents cumulative discounted rewards from time

t to the episode's end. This does use bootstrapping: it does not estimate value using existing estimates like $V(s')$ or $Q(s', a')$. Instead, it waits for the full return to be observed before making updates, making the updates unbiased. For episodic tasks with exploring starts, Monte Carlo methods converge to the optimal policy as the number of episodes approaches infinity Tsitsiklis (2002).

Lastly, we touch on policy-gradient methods. REINFORCE directly optimizes the policy parameters θ based on observed episode returns G, following the gradient direction that increases the probability of actions leading to higher rewards. The policy parameterized by, typically, a neural network:

$$\text{REINFORCE:} \quad \theta \leftarrow \theta + \alpha \nabla_\theta \log \pi_\theta(a|s) G \tag{7}$$

One can interpret the REINFORCE as trying to generate data from a policy and re-estimating the policy parameters using maximum likelihood via gradient ascent, but with increased weight when the trajectory returns larger returns. This step pulls the policy towards generating larger returns. REINFORCE converges to a stationary point of the expected return function with probability 1 (Williams, 1992b). In tabular settings with suitable exploration, it achieves global optimality at a geometric rate Agarwal et al. (2020).

3.1 Example 1: Grid Navigation

To illustrate the connection between DP and RL methods, we consider a 5×5 grid world navigation problem forming a finite MDP where states are coordinates (r, c) with $0 \leq r, c < 5$; actions are directional movements {L, R, U, D, S}[1]; transitions follow deterministic movement rules with boundary constraints; rewards include +10 at the terminal state $(4, 4)$, smaller intermediate rewards throughout the grid, and a movement penalty of -0.1; and the discount factor γ equals 0.95. Table 1 presents solutions from all tabular methods.

[1]L=Left, R=Right, U=Up, D=Down, S=Stay

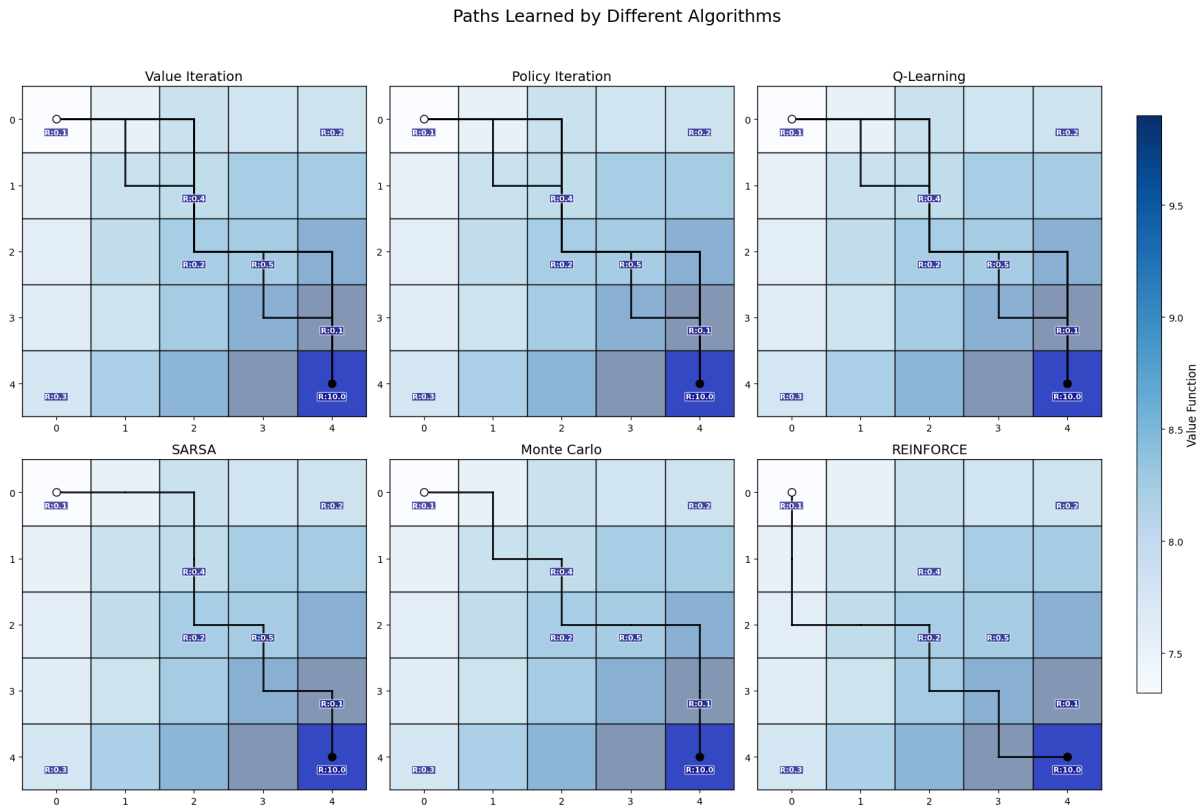


Figure 1: Paths found by each algorithm.

Table 1: Performance comparison of DP and RL methods on the grid world problem[2]

Method	**MSE***	**Policy Match***	**Time (s)**	**NumIter**	**Episodes***	**Return***
VI	0.00	1.00	0.00	9	8.00	10.40
PI	0.00	1.00	0.01	7	8.00	10.40
Q-Learning	0.00	1.00	0.04	1211	8.00	10.40
SARSA	1.57	0.92	0.08	5000	8.00	10.40
Monte Carlo	0.12	0.88	0.09	5000	8.00	10.40
REINFORCE	72.94	0.75	12.97	5008	8.00	9.40

VI and PI converge exactly and rapidly to optimal solutions with perfect policy matches, leveraging their complete knowledge of the MDP. Q-learning achieves perfect policy matching despite lacking prior model knowledge, though requiring more iterations to converge. SARSA and Monte Carlo find valid paths to the goal but do not identify all optimal actions, instead converging to a single path with reasonable value approximation. As shown in Figure 1, these RL methods favor consistent exploitation once a workable solution is found. REINFORCE shows the most variable performance, achieving only 75% policy matching and suboptimal rewards, highlighting the sensitivity of policy gradient methods to hyperparameter tuning and neural network initialization. This example

illustrates that while model-free RL methods can solve tasks without prior knowledge of transition and reward functions, they typically require more samples and tuning and may not identify all optimal actions that DP methods discover.

3.2 Deep Reinforcement Learning

Deep Q-Network (DQN) extends Q-learning by approximating the action-value function with a deep neural network $Q(s, a; \theta)$. Instead of storing values in a table, DQN uses a parameterized model trained on transition samples (s, a, r, s') drawn from a replay buffer. Its objective is to minimize the temporal-difference (TD) error:

$$L(\theta) \;=\; \mathbb{E}_{(s,a,r,s')}\Big[\Big(r \;+\; \gamma \max_{a'} Q(s', a'; \theta^{-}) \;-\; Q(s, a; \theta)\Big)^2\Big].$$

Here, θ^{-} denotes the parameters of a separate target network, periodically synchronized to prevent instability from moving targets. By replaying past transitions and using a target network, DQN decouples correlated updates and stabilizes learning, enabling it to scale to high-dimensional state spaces. Under proper conditions including Lipschitz-continuous Q-function and sufficient exploration, DQN converges to an approximate Bellman fixed point Ramaswamy (2021). With experience replay and target networks breaking the "deadly triad" instability, DQN can achieve geometric convergence to Q^* when the network class can represent it Zhang et al. (2023).

3.3 Example 2: Capital Replacement

We study a maintenance optimization problem with N parallel engines formulated as a Markov Decision Process. Each engine i has mileage state $m_i \in 0, 1, 2, 3, 4, 5$, with system state $s = (m_1, ..., m_N)$. The action space $A(s)$ consists of all subsets of engines to replace, constrained by capacity $C = 3$. The cost function is:

$$c(s, a) = \alpha \sum_{i=1}^{N} \mathbf{1} m_i > 0 + \beta |a| \tag{8}$$

where $\alpha = 1.0$ is the cost of aged engines, $\beta = 5.0$ is the replacement cost, and $|a|$ is the number of replaced engines. State transition is deterministic:

$$m_i' = \begin{cases} 0 & \text{if } i \in a \ \ \min(5, m_i + 1) \\ \text{otherwise} \end{cases} \tag{9}$$

The objective is to find policy π minimizing expected discounted cost with $\gamma = 0.95$:

$$V^(s) = \min \pi \mathbb{E}\pi \left[\sum t = 0^{\infty} \gamma^t c(s_t, \pi(s_t)) \right] \tag{10}$$

For the DQN implementation, we map the system state $s = (m_1, \ldots, m_N)$ directly to the neural network input[3], where each mileage value m_i becomes a feature in the input vector. The action space is encoded as indices of a fixed-size output vector, where index j corresponds to the j-th possible subset of engines to replace (e.g., empty set, single engines, pairs, and triples up to capacity $C = 3$).

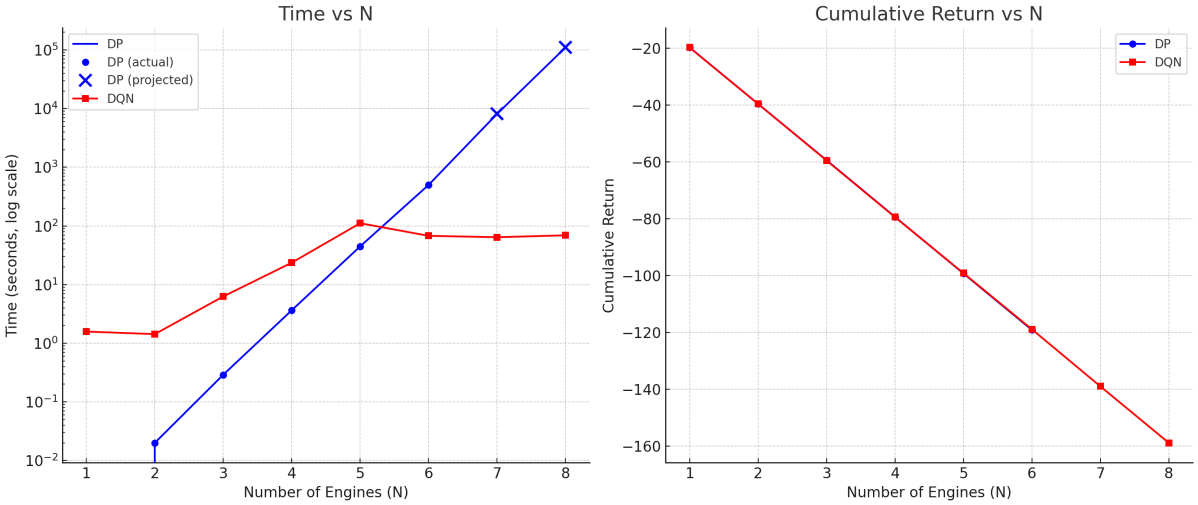
Dynamic Programming (DP) solves this exactly via value iteration, but faces exponential state growth $\mathcal{O}(6^N)$. For each state, DP evaluates all possible actions and selects the one minimizing the Bellman equation until convergence, measured by maximum value change between iterations falling below 1e-5. DQN addresses the scalability challenge by approximating the value function and learning from sampled experiences, with convergence measured through the Bellman residual—the average absolute difference between the current Q-value estimate and the target.

N	#States	DP Time	DP Reward	DQN Time	DQN Reward	DQN Bell Res
1	6	0.00s	-19.80	1.58s	-19.83	0.44
2	36	0.02s	-39.64	1.43s	-39.63	0.86
3	216	0.29s	-59.42	6.27s	-59.53	0.79
4	1,296	3.64s	-79.35	23.60s	-79.43	2.13
5	7,776	44.82s	-99.26	111.33s	-99.15	1.82
6	46,656	500.98s	-119.10	67.81s	-118.88	2.79
7	279,936	∼6418.84s	–	64.18s	-138.93	2.58
8	1,679,616	∼78561.36s	–	69.25s	-158.80	2.92

For small instances ($N \leq 3$), dynamic programming (DP) yields exact solutions in

[3]The neural network architecture consists of fully-connected layers (128-64) with ReLU activations, trained using Adam optimizer with learning rate 1e-3 and batch size 64. To ensure stability, we employ ϵ-greedy exploration starting at $\epsilon = 1.0$ and decaying by a factor of 0.995 per episode to a minimum of 0.01, along with a target network updated every 50 episodes.

under 1 second, while deep Q-networks (DQN) require 20–30 seconds to achieve comparable performance. As the state space grows, DP becomes intractable—solving $N = 6$ (46,656 states) takes over 500 seconds, and projections for $N = 7$ exceed 6,000 seconds. In contrast, DQN scales efficiently and maintains high solution quality, with rewards closely matching those from DP wherever comparison is possible.



3.4 Performance of Reinforcement Learning on Optimization Problems

Multiple empirical studies provide evidence that Deep RL can indeed overcome the curse of dimensionality that limits traditional Dynamic Programming. Nomura et al. (2025) showed that for perishable inventory control with dynamic pricing, PPO achieved near-optimal solutions with a 75% reduction in computation time compared to DP, with authors explicitly noting that increasing problem dimensionality would render DP impractical while DRL would remain viable. Sabri et al. (2024) demonstrated that attention-based DRL delivered maintenance schedules with 30.5% lower costs than DP-based heuristics on combinatorial problems, computing in seconds what DP required over an hour to solve. For high-dimensional inventory problems where exact DP was infeasible, Gijsbrechts et al. (2022) showed that A3C matched the performance of the best specialized heuristics and approximate DP methods across three inventory settings. These preliminary results show that while Deep RL requires substantial initial training experiences and hyperparameter tuning, they can effectively approximate optimal policies in high-dimensional state spaces where exact DP becomes computationally prohibitive.

4 Reinforcement Learning for Structural Estimation

This section reviews reinforcement learning methods to solve structural economic models.

4.1 Experience-Based Equilibria

Fershtman and Pakes (2012) introduced Experience-Based Equilibrium (EBE) for dynamic games with asymmetric information. The EBE concept defines equilibrium as a triple $(S^*, \{\pi_i\}, \{V_i\})$ consisting of recurrent states, strategies, and value functions satisfying recurrence, optimality, and consistency conditions.

The computational algorithm involves simulating industry trajectories, updating strategies to best responses, and adjusting value estimates based on observed outcomes. This approach evaluates values only on recurrently visited states rather than the full state space, avoiding integration over unobserved state distributions.

Asker et al. (2020) analyze pricing strategies in a Bertrand duopoly where firms repeatedly set prices for a homogeneous good. Each firm maintains a value estimate $Q(s, a)$ for state-action pairs that follows the Bellman equation:

$$Q(s, a) = \mathbb{E}[r(s, a) + \beta \max_{a'} Q(s', a')] \tag{11}$$

After each pricing decision, firms update their Q-values using:

$$Q_i(s, a) \leftarrow Q_i(s, a) + \alpha[\pi_i + \beta \max_{a'} Q_i(s', a') - Q_i(s, a)] \tag{12}$$

where α is the learning rate and β the discount factor. The authors find certain learning protocols lead to supracompetitive pricing, while others converge to competitive outcomes.

Building on the EBE concept, Lomys and Magnolfi (2024) develop a structural estimation approach for strategic settings where agents use reinforcement learning rather than playing a fixed equilibrium. They impose an "asymptotic no-regret" (ANR) condition as a minimal rationality requirement, showing that the empirical distribution of play must satisfy certain Bayes correlated equilibrium conditions. Using these restrictions, they identify and estimate underlying payoff parameters of repeated games played by learning agents, providing dynamic foundations for structural econometric inference in

multi-agent systems without assuming Nash equilibrium.

4.2 Approximating Auction Equilibria and Mechanism Design

Graf et al. (2025) evaluate deep RL in finding equilibria for sealed-bid auctions using Deep Deterministic Policy Gradient (DDPG). Through extensive hyperparameter tuning, they achieve 99% convergence to the theoretical Bertrand-Nash equilibrium, providing rigorous validation that deep RL can find provable economic equilibria.

Brero et al. (2021) pioneer the use of RL to design economic mechanisms, focusing on Sequential Price Mechanisms (SPM) – a class of indirect auction mechanisms where agents are approached in sequence and offered choice menus at posted prices. They formulate the mechanism design problem as a partially observed MDP: the "state" includes the history of past sales, and the mechanism's policy chooses prices and allocation order based on this history. Using deep RL (policy optimization), they learn optimal or near-optimal sequential pricing policies in various simulated environments, outperforming static pricing benchmarks. They also provide theoretical conditions for when an adaptive sequential mechanism can yield higher welfare or revenue than simpler static mechanisms.

Ravindranath et al. (2024) address the challenge of learning revenue-maximizing mechanisms for sequential combinatorial auctions involving multiple items and strategic bidders. They propose a specialized RL framework that leverages differentiable structure in the auction's transitions to enable gradient-based learning. By exploiting the fact that auction clearing and payment rules can be made differentiable, they integrate first-order optimization into the agent's policy update. Their results show significant improvements in revenue over both analytic benchmarks and conventional deep RL algorithms. Their approach scales to environments with up to 50 bidders and 50 items, effectively bridging the gap between auction theory's optimal mechanisms and practical implementable policies.

4.3 Dynamic Oligopoly Models

Hollenbeck (2019) applies reinforcement learning to analyze merger effects on innovation in dynamic oligopoly. Firms update value estimates through temporal-difference learning:

$$V_{\text{new}}(s) \leftarrow V(s) + \alpha[\pi(s,a) + \beta V(s') - V(s)] \tag{13}$$

The algorithm employs ϵ-greedy exploration to prevent convergence to suboptimal equilibria. The RL approach handles high-dimensional state spaces that include multiple firms and quality levels, making traditional dynamic programming infeasible.

Covarrubias et al. (2022) uses deep reinforcement learning to study oligopolistic pricing in a New Keynesian framework. Neural networks represent firms' pricing policies $\pi_\theta(a|s)$ and value functions, enabling firms to learn sophisticated strategies including history-dependent "trigger" strategies supporting tacit collusion. The method uncovers multiple equilibria ranging from competitive to collusive pricing with higher markups.

4.4 Dynamic Discrete Choice Models

Hu and Yang (2025) develop an estimator combining policy gradient methods with indirect inference for dynamic discrete choice models. Their approach employs a two-loop structure where the outer loop matches moments:

$$\hat{\theta} = \arg\min_{\theta} \left(\mathbf{M}_d - \mathbf{M}_s(\theta)\right)' \mathbf{W} \left(\mathbf{M}_d - \mathbf{M}_s(\theta)\right) \tag{14}$$

The inner loop solves the dynamic programming problem via policy gradient:

$$\nabla_\gamma J(\gamma) = \mathbb{E}_{\tau\sim\pi_\gamma}\left[\sum_{t=0}^{T} \nabla_\gamma \log \pi_\gamma(a_t|s_t) Q^\pi(s_t, a_t)\right] \tag{15}$$

This direct policy parameterization sidesteps integration over continuous state transitions that burdens traditional nested fixed-point estimation. The policy is modeled as:

$$\pi(a_t|s_t, \eta_t; \gamma) = \frac{\exp\left(f_\gamma(s_t, \eta_t, a_t)\right)}{\sum_{j\in\mathcal{A}} \exp\left(f_\gamma(s_t, \eta_t, j)\right)} \tag{16}$$

where η_t denotes unobserved states and f_γ is implemented as a neural network.

Adusumilli et al. (2022) introduce temporal-difference (TD) learning methods to estimate structural parameters of dynamic discrete choice models. They adapt the econometric conditional choice probability approach to an RL-style algorithm, using function approximation to avoid explicit transition probability specification. Their two estimators (a linear TD method and an approximate value iteration) can handle continuous or high-dimensional state spaces and even multi-agent dynamic games without integrating over other players' actions. Monte Carlo experiments demonstrate these RL-based estimators

are computationally fast and statistically consistent.

4.5 Macroeconomic Models

Atashbar and Shi (2023) apply Deep Deterministic Policy Gradient to solve a Real Business Cycle model. They implement two neural networks: an actor $\mu_\theta(s)$ representing policy and a critic $Q_\phi(s, a)$ estimating action values. The critic network minimizes:

$$\mathcal{L}(\phi) = \mathbb{E}_{(s,a,r,s')\sim\mathcal{D}}\left[\left(Q_\phi(s,a) - (r + \gamma Q_{\phi'}(s', \mu_{\theta'}(s')))\right)^2\right] \tag{17}$$

The actor network updates via policy gradient:

$$\nabla_\theta J(\theta) = \mathbb{E}_{s\sim\mathcal{D}}\left[\nabla_a Q_\phi(s,a)\Big|_{a=\mu_\theta(s)} \nabla_\theta \mu_\theta(s)\right] \tag{18}$$

Their learned policies closely approximate theoretical optimal policies without requiring explicit derivation of Euler equations.

Curry et al. (2022) employ multi-agent reinforcement learning to compute equilibria in heterogeneous agent macro models with up to 111 agents. They introduce structured learning curricula to stabilize training, gradually scaling from simplified economies to full models.

Hinterlang and Taenzer (2024) combine empirical model estimation with RL-based monetary policy optimization. They first estimate a neural network transition model, then optimize central bank policy to minimize:

$$\mathcal{L}_{\text{policy}} = \mathbb{E}\left[\sum_t \lambda_\pi(\pi_t - \pi^*)^2 + \lambda_y(y_t - y^*)^2\right] \tag{19}$$

Their approach discovers nonlinear policy rules outperforming conventional Taylor rules by 27-43%, demonstrating RL's ability to derive sophisticated policies in dynamic environments.

4.6 Advantages and Limitations

Reinforcement learning offers several key advantages for structural economic models. Firstly, RL methods with function approximation tackle high-dimensional state spaces that render traditional dynamic programming infeasible. Neural network representations

can handle continuous state variables and state dependencies without discretization, as demonstrated by Covarrubias et al. (2022) and Curry et al. (2022). Secondly, by focusing computational resources on recurrently visited states rather than exhaustively solving for all possible states, RL methods achieve significant efficiency gains. Fershtman and Pakes (2012) and Hollenbeck (2019) leverage this property to solve models with many firms and state variables. Thirdly, experience-based equilibrium concepts integrated with RL provide tractable solutions for settings with private information where standard methods fail. This allows modeling strategic interactions without requiring explicit integration over belief states. Lastly, the stochastic nature of RL exploration helps uncover multiple equilibria in games. Covarrubias et al. (2022) demonstrates this by finding both competitive and collusive equilibria in the same model setup.

Despite these strengths, RL methods face important limitations. Firstly, even single-agent deep RL lacks general theoretical convergence guarantees in economic applications. While empirical convergence can be validated through testing potential improvements (-equilibrium), verification remains computationally intensive. Secondly, MARL may fail to detect unstable equilibria, often oscillates during training, and can miss certain equilibria entirely. Learning dynamics are sensitive to initialization and equilibrium selection is not fully understood. Lastly, These methods remain computationally demanding, requiring GPU acceleration and careful hyperparameter tuning. As shown by Graf et al. (2025), achieving reliable convergence requires systematic optimization of training parameters.

5 Inverse Reinforcement Learning

5.1 Key IRL Methodologies

Inverse Reinforcement Learning (IRL) addresses a fundamental problem: given observed agent behavior, how can we recover the reward function that motivated it? Formally, given an MDP $\mathcal{M} \setminus R = \{S, A, P, \gamma\}$ without the reward function, and demonstrations $\mathcal{D} = \{\tau_1, \tau_2, ..., \tau_m\}$ where each trajectory $\tau_i = \{(s_0, a_0), (s_1, a_1), ..., (s_T, a_T)\}$ consists of state-action pairs, the goal is to recover the reward function $R : S \times A \to \mathbb{R}$ that the demonstrator is optimizing (Ng and Russell, 2000).

5.1.1 Max-Margin IRL

Ng and Russell (2000) introduced the max-margin approach, establishing that a reward function should make the expert's policy better than alternatives by some margin. For all states $s \in S$ and actions $a \neq \pi_E(s)$, it enforces constraints:

$$(P_{\pi_E(s)} - P_a)(I - \gamma P_{\pi_E})^{-1} R \geq 0$$

where P_a is the transition matrix for action a, P_{π_E} is the transition matrix under the expert policy, γ is the discount factor, and R is the reward vector. The objective maximizes this margin while keeping rewards bounded:

$$\max_R \sum_{s \in S} \min_{a \neq \pi_E(s)} [(P_{\pi_E(s)} - P_a)(I - \gamma P_{\pi_E})^{-1} R] - \lambda ||R||_1$$

where λ controls the reward sparsity. Ratliff et al. (2006) extended this approach with Maximum Margin Planning, formulating a quadratic program that makes the expert's behavior optimal under the learned reward. This approach requires known transition dynamics and assumes expert optimality, limiting its applicability in environments with uncertain dynamics (Abbeel and Ng, 2004).

5.1.2 Maximum Entropy IRL

Ziebart et al. (2008) introduced a probabilistic approach using the principle of maximum entropy to resolve ambiguity in IRL. Rather than assuming deterministic optimality, MaxEnt IRL models the expert's trajectory distribution as:

$$P(\tau|\theta) = \frac{1}{Z(\theta)} \exp(\theta^T \sum_{t=0}^{T} \phi(s_t))$$

where τ is a trajectory, θ are reward parameters, $\phi(s_t)$ are state features, and $Z(\theta)$ is the partition function. This formulation is equivalent to assuming the agent follows a Boltzmann policy, mathematically corresponding to a dynamic discrete choice model with logit errors (Ermon et al., 2015). The objective becomes maximum likelihood estimation of reward parameters, solved via gradient ascent:

$$\nabla L(\theta) = \tilde{\phi} - \sum_s D_s \phi_s$$

where $\tilde{\phi}$ represents empirical feature counts from demonstrations and D_s are expected state visitation frequencies. Ziebart et al. (2008) demonstrated this approach's effectiveness for predicting pedestrian movement, achieving 74.8% accuracy compared to 45.9% for shortest-path predictions. Wulfmeier et al. (2015) later extended this to deep neural network reward representations, enabling applications in environments with high-dimensional features.

5.1.3 Generative Adversarial Imitation Learning

Ho and Ermon (2016) reframed IRL as a distribution matching problem using adversarial training. GAIL employs a discriminator network $D_\omega : S \times A \to [0, 1]$ to distinguish between expert and learner behaviors, while a policy network $\pi_\theta : S \to P(A)$ tries to fool the discriminator:

$$\min_\theta \max_\omega \mathbb{E}_{\tau\sim\pi_\theta}[\log(1 - D_\omega(s, a))] + \mathbb{E}_{\tau\sim\pi_E}[\log(D_\omega(s, a))] - \lambda H(\pi_\theta)$$

where $H(\pi_\theta)$ is the entropy regularization term. The discriminator effectively learns a reward function: $\log(D_\omega(s, a)) - \log(1 - D_\omega(s, a))$. Fu et al. (2018) extended this to Adversarial IRL, which explicitly recovers a reward function that generalizes beyond training trajectories, demonstrating improved transfer to modified environments in robotic tasks. Kostrikov et al. (2018) further refined this approach with Discriminator Actor-Critic, improving sample efficiency through off-policy training.

Summary of Identifiability Conditions in IRL. Inverse Reinforcement Learning (IRL) is generally underdetermined, because multiple reward functions can induce the same observed policy (Kim et al., 2021). For instance, any constant shift on the reward is unidentifiable. Nonetheless, recent theoretical work has established conditions for unique (up to a constant) identifiability. First, if we assume deterministic MDPs combined with maximum-entropy regularization, observed trajectory frequencies directly reveal the reward (Kim et al., 2021). In more general settings, coverage assumptions ensure every state-action lies on some optimal trajectory, which provides enough constraints for uniqueness (Kim et al., 2021). Another key line of work uses multiple environments (or discount factors): consistent optimal behavior in diverse MDPs forces a single reward function across them all (Cao et al., 2021; Rolland et al., 2022b). If the reward is lin-

ear in known features and the constant function is excluded from the feature span, then the parameter vector becomes uniquely identifiable (Rolland et al., 2022a; Shehab et al., 2024). Finally, simplifying the reward structure—for example, restricting to state-only or time-homogeneous rewards—removes degrees of freedom and thus promotes uniqueness (Cao et al., 2021). Overall, these approaches illustrate how additional assumptions or data sources can make IRL well-posed, with a single consistent reward function (up to additive constants).

5.2 Identification Challenges and Econometric Connections

Different IRL methods address identification through various mechanisms. Max-Margin IRL selects the reward function with maximum margin between optimal and suboptimal actions (Ratliff et al., 2006). MaxEnt IRL applies maximum entropy as a selection principle, providing a unique solution consistent with observed behavior frequencies (Ziebart et al., 2008). Kim et al. (2021) formalized necessary and sufficient conditions for reward identifiability, proving that specific properties of the environment and policy determine whether rewards can be uniquely recovered.

O'Briant (2024) demonstrates that standard IRL's use of "wind shocks" yields only partial identification, because random deviations cannot reveal relative payoffs. Substituting preference shocks from the DDC framework of Rust (1987) restores full identification. Under this assumption, Abbeel and Ng (2004)'s projection IRL attains nearly the accuracy of NFXP but requires up to 20× fewer dynamic programming solves. By contrast, the linear program of Ng and Russell (2000) remains computationally simpler yet restricts payoffs to $\{-1, 0, 1\}$.

The connection between IRL and structural econometrics is mathematically precise. MaxEnt IRL's probabilistic formulation is equivalent to a dynamic discrete choice model with logit errors (Ermon et al., 2015). The Boltzmann policy corresponds to assuming i.i.d. Gumbel noise in utilities, leading to choice probabilities matching multinomial logit specifications. This equivalence was validated by Ermon et al. (2015), who showed that MaxEnt IRL solutions coincide with maximum-likelihood estimates from corresponding structural models.

Sharma et al. (2018) operationalized this connection by adapting Hotz-Miller's conditional choice probability (CCP) approach from econometrics to IRL. Their CCP-IRL

approach achieved a 5× speed improvement with no loss in reward quality compared to standard MaxEnt IRL on benchmark tasks. This computational advance addresses a key challenge in both fields: the nested fixed-point computation required in each iteration. In IRL, each reward update typically requires solving a forward MDP—analogous to the nested fixed-point algorithms in structural estimation (Rust, 1987).

Zeng et al. (2022) proposed a single-loop algorithm that interleaves policy optimization with reward parameter updates, proving convergence to stationary points with theoretical guarantees. This approach parallels simultaneous estimation methods in econometrics, both avoiding full policy convergence at each iteration. Boularias et al. (2011) developed Relative Entropy IRL, a model-free approach using KL-divergence that avoids requiring known transition dynamics, making it applicable to settings where the environment model is unavailable.

5.3 Economic Applications of IRL

IRL methods have been applied to domains that resemble economic decision-making, demonstrating potential to complement structural econometric modeling in environments.

5.3.1 Dynamic Discrete Choice and Migration Decisions

Ermon et al. (2015) applied IRL to model pastoralist farmers' movement decisions in East Africa using GPS-tracked movements. Their study treated migration routes as optimal policies in an MDP where rewards reflected preferences for resource abundance versus travel costs. The model quantified trade-offs between water availability, grazing quality, and distance traveled. The authors found that drought conditions would significantly alter movement patterns, with herders traveling 21% farther on average to reach water sources—a finding with implications for climate adaptation policy.

This application highlights how IRL can handle high-dimensional state spaces and dynamics that would challenge traditional structural estimation. Bogota et al. (2015) applied similar techniques to refugee migration patterns, demonstrating how IRL can recover preference parameters from observational data even when underlying utility functions are non-linear.

5.3.2 Route Choice and Transportation Economics

IRL has been used to infer driver preferences in route selection, complementing discrete choice models in transportation economics. Ziebart et al. (2008) applied MaxEnt IRL to predict taxi navigation paths, revealing that drivers optimize reward functions beyond simple distance minimization. Bronner et al. (2023) utilized IRL to detect anomalous ride-hailing driver routes, identifying preferences for specific road types and time-dependent routing strategies with 89% accuracy.

Mai et al. (2015) compared IRL approaches to traditional route choice models, finding that IRL captured path dependencies and strategic anticipation that multinomial logit models missed. Their analysis showed that IRL-derived models reduced prediction error by 17% when forecasting traffic redistribution after infrastructure changes, highlighting IRL's value for transportation policy evaluation.

5.3.3 Labor Market and Skill Development

Yancey et al. (2022) applied IRL to model worker skill development and career path choices, treating labor market participation as a sequential decision problem. By analyzing career trajectory data from 15,000 workers over a ten-year period, the study estimated that workers valued skill development opportunities at approximately 14-18% of immediate compensation, varying by education level. The resulting model predicted responses to unemployment benefit changes with higher accuracy than traditional structural labor models.

Chan et al. (2019) extended this approach to incorporate bounded rationality in job search, showing how IRL can recover discount factors and risk preferences simultaneously with reward functions. Their findings indicated significant heterogeneity in how workers value job security versus wage growth potential, with implications for designing effective unemployment insurance programs.

These applications demonstrate IRL's ability to recover preference parameters from observed behavior in dynamic settings, enabling counterfactual analysis and policy evaluation—core objectives of structural economic analysis. The field continues to develop, with recent work by Ramachandran and Amir (2007) and Finn et al. (2016) introducing Bayesian IRL and Guided Cost Learning approaches that further enhance IRL's applica-

bility to economic modeling.

6 Reinforcement Learning in Games

Multi-agent reinforcement learning (MARL) operates at the intersection of traditional reinforcement learning and game theory. While single-agent RL concerns optimal behavior in stochastic environments, MARL extends this paradigm to settings where multiple agents simultaneously learn and adapt, naturally giving rise to strategic interactions analyzed in game theory.

6.1 Replicator Dynamics for Reinforcement Learning

We examine three key reinforcement learning algorithms and analyze their connection to evolutionary game theory of normal-form games through the lens of dynamical systems.

6.1.1 Cross Learning

Cross-learning (Cross, 1973) represents one of the earliest reinforcement learning algorithms with direct connections to evolutionary dynamics. The discrete-time update rule for action probabilities is given by:

$$\pi(i) \leftarrow \pi(i) + \begin{cases} r(1 - \pi(i)) & \text{if } i = j \\ -r\pi(i) & \text{if } i \neq j \end{cases} \tag{20}$$

where j is the selected action and $r \in [0, 1]$ is the received reward.

Börgers and Sarin (1997) showed that as the learning rate becomes infinitesimal, the expected behavior of Cross learning converges to:

$$\dot{\pi}(i) = \pi(i) \left[\mathbb{E}[r(i)] - \sum_j \pi(j)\mathbb{E}[r(j)] \right] \tag{21}$$

This equation precisely matches the canonical replicator dynamics from evolutionary game theory:

$$\dot{x}_i = x_i(f_i(x) - \bar{f}(x)) \tag{22}$$

where x_i represents the probability of action i, $f_i(x)$ is the fitness (expected reward) of

action i, and $\bar{f}(x) = \sum_j x_j f_j(x)$ is the average fitness across all actions. This equivalence establishes an important link: individual learning through Cross's rule produces population-level dynamics identical to biological evolution under replicator dynamics. In game-theoretic terms, strategies with above-average payoffs increase in frequency, while below-average strategies decline.

6.1.2 Q-learning with Boltzmann Exploration

The standard Q-learning update rule is given by:

$$Q(s,a) \leftarrow Q(s,a) + \alpha \left[r + \gamma \max_{a'} Q(s',a') - Q(s,a) \right] \tag{23}$$

When combined with Boltzmann exploration:

$$\pi(a) = \frac{e^{Q(a)/\tau}}{\sum_j e^{Q(j)/\tau}} \tag{24}$$

where τ is the temperature parameter controlling exploration. Tuyls et al. (2003) demonstrated that in stateless games, the dynamics of Q-learning with Boltzmann exploration can be approximated by:

$$\dot{x}_i = \alpha x_i \left[\frac{(Ay)_i - x^\top Ay}{\tau} - \log x_i + \sum_j x_j \log x_j \right] \tag{25}$$

where A is the game payoff matrix, y is the opponent's strategy, and x_i is the probability of action i. This equation decomposes into two components, (1) an exploitation term resembling replicator dynamics, scaled by $\frac{1}{\tau}$ and (2) an exploration term derived from the entropy gradient. As $\tau \to 0$ (pure exploitation), the dynamics converge to standard replicator dynamics. As $\tau \to \infty$ (pure exploration), the system moves toward uniform randomization.

6.1.3 Policy Gradient Methods

Policy gradient methods directly optimize expected rewards through gradient ascent on policy parameters:

$$\pi \leftarrow \pi + \alpha \nabla_\pi \mathbb{E}[R] \tag{26}$$

For two-action games, with $x \in [0, 1]$ representing the probability of the first action, the expected reward is:

$$V(x) = x(Ay)_1 + (1 - x)(Ay)_2 \tag{27}$$

The Infinitesimal Gradient Ascent (IGA) dynamics become:

$$\dot{x} = \alpha \frac{dV}{dx} = \alpha((Ay)_1 - (Ay)_2) \tag{28}$$

Unlike Cross learning and Q-learning with Boltzmann exploration, policy gradient methods yield linear dynamics rather than replicator-like equations. Notably, the update does not depend on $x(1 - x)$, lacking the characteristic sigmoidal shape of replicator dynamics. IGA can converge to interior Nash equilibria in zero-sum games (Singh et al., 2000) but may exhibit cycles in general-sum games. Extensions like Win-or-Learn-Fast (WoLF) policy hill-climbing (Bowling and Veloso, 2002) introduce adaptive learning rates to overcome cycling, with higher learning rates when "losing" and lower rates when "winning."

6.2 Learning Algorithms in Competitive Multi-Agent Settings

We now examine key algorithm classes for competitive multi-agent environments, focusing on value-based, policy-based, and extensive-form game approaches. Each class comes with distinct theoretical properties and equilibrium guarantees, particularly relevant to adversarial settings prevalent in economics.

6.2.1 Value-Based Learning

Value-based methods learn action-value functions to determine optimal policies, with varying approaches to handling opponent behavior.

Independent Q-Learning (IQL). The simplest approach treats other agents as part of the environment, with each agent applying standard Q-learning:

$$Q(s, a) \leftarrow Q(s, a) + \alpha[r + \gamma \max_{a'} Q(s', a') - Q(s, a)] \tag{29}$$

While computationally efficient, IQL suffers from fundamental convergence issues in

multi-agent settings. The environment becomes non-stationary from each agent's perspective as other agents simultaneously learn, violating the Markov property required for Q-learning's convergence guarantees (Tan, 1993). In competitive environments, this approach typically fails to find equilibrium strategies.

Minimax Q-Learning. Littman (1994) developed Minimax Q-learning specifically for two-player zero-sum Markov games. Instead of the optimistic maximization in standard Q-learning, it uses the minimax value:

$$V(s) = \max_{\pi \in \Delta(A)} \min_{b} \sum_{a} \pi(a) Q(s, a, b) \tag{30}$$

$$Q(s, a, b) \leftarrow Q(s, a, b) + \alpha[r + \gamma V(s') - Q(s, a, b)] \tag{31}$$

This algorithm guarantees convergence to minimax equilibrium values under standard assumptions of sufficient exploration and learning rate decay. It requires solving a linear program at each state, becoming computationally intensive as the action space grows.

Nash Q-Learning. Hu and Wellman (2003) extended Minimax Q-learning to general-sum games. At each state, it computes the Nash equilibrium of the stage game:

$$Q^i(s, \mathbf{a}) \leftarrow Q^i(s, \mathbf{a}) + \alpha[r^i + \gamma \mathbb{E}_{\pi^*}[Q^i(s', \cdot)] - Q^i(s, \mathbf{a})] \tag{32}$$

where π^* represents the Nash equilibrium distribution.

Nash Q-learning proves convergence under restrictive assumptions, including the uniqueness of equilibrium at each state. The computational complexity of repeatedly solving for Nash equilibria (NP-hard for general-sum games) limits its practical application beyond small games (Daskalakis et al., 2009).

6.2.2 Policy-Based Learning

Policy-based methods optimize parameterized policies directly via gradient ascent on expected returns, offering better scalability in large state spaces.

Policy Gradient Methods. In multi-agent settings, each agent i applies gradient ascent on its policy parameters:

$$\theta_i \leftarrow \theta_i + \alpha \nabla_{\theta_i} J_i(\theta_i) \tag{33}$$

where $J_i(\theta_i)$ is agent i's expected return. The policy gradient theorem (Sutton et al., 1999a) gives:

$$\nabla_{\theta_i} J_i(\theta_i) = \mathbb{E}_{\pi_i}[\nabla_{\theta_i} \log \pi_i(a_i|s) \cdot Q^{\pi_i}(s, a_i)] \tag{34}$$

In competitive settings, when agents independently apply policy gradients, convergence to Nash equilibria is not guaranteed. Mazumdar and et al. (2019) demonstrated that even in simple differentiable games, standard policy gradient updates can become trapped in limit cycles around equilibria.

WoLF Policy Gradient. The Win or Learn Fast principle (Bowling and Veloso, 2002) introduces an adaptive learning rate mechanism:

$$\alpha = \begin{cases} \alpha_{\text{win}}, & \text{if current policy outperforms reference policy} \\ \alpha_{\text{lose}}, & \text{otherwise} \end{cases} \tag{35}$$

where $\alpha_{\text{win}} < \alpha_{\text{lose}}$

By adjusting learning rates based on performance relative to a reference policy (often a historical average), WoLF stabilizes learning dynamics. It has proven convergence properties in two-player, two-action games and empirically improves convergence in larger settings. The key insight is to slow down learning when winning to avoid destabilizing emerging equilibria.

6.2.3 Extensive-Form Game Algorithms

Designed for sequential, partially-observable games, these methods handle information asymmetry and sequential decision-making.

Fictitious Play. In classical fictitious play (Brown, 1951), each agent maintains beliefs about opponents based on historical actions and best-responds accordingly:

$$\pi_i^{(t+1)} = \text{BR}(\bar{\pi}_{-i}^{(t)}) \tag{36}$$

where $\bar{\pi}_{-i}^{(t)}$ is the empirical average of opponent strategies up to round t.

Fictitious play converges to Nash equilibria in zero-sum games (Robinson, 1951), potential games (Monderer and Shapley, 1996), and other restricted classes, but Shapley (1964) demonstrated cycling behavior in certain games.

Neural Fictitious Self-Play (NFSP). Heinrich and Silver (2016) combined fictitious play with deep learning, enabling scalability to larger games. NFSP maintains two networks for each player: a best-response policy β_i trained via deep Q-learning and an average strategy network $\bar{\pi}_i$ trained via supervised learning. The best-response network is updated according to:

$$L_i^{\text{RL}}(\theta_i) = \mathbb{E}_{(s,a_i,r_i,s')\sim\mathcal{M}_i^{\text{RL}}}[(r_i + \gamma \max_{a_i'} Q_i(s', a_i'; \theta_i^-) - Q_i(s, a_i; \theta_i))^2] \tag{37}$$

where θ_i^- are the parameters of a target network. The average strategy network is trained via supervised learning on best-response actions:

$$L_i^{\text{SL}}(\phi_i) = \mathbb{E}_{(s,a_i)\sim\mathcal{M}_i^{\text{SL}}}[-\log \bar{\pi}_i(a_i|s;\phi_i)] \tag{38}$$

where $\mathcal{M}_i^{\text{SL}}$ is a reservoir buffer of best-response actions. NFSP has demonstrated convergence to approximate Nash equilibria in games like Leduc poker, though subsequent methods have shown stronger performance in larger games.

Counterfactual Regret Minimization (CFR). Zinkevich et al. (2008) developed CFR for extensive-form games, tracking regret at each information set:

$$R^T(I, a) = \sum_{t=1}^{T} [v^t(I, a) - v^t(I)] \tag{39}$$

where $v^t(I, a)$ is the counterfactual value of action a at information set I.

The policy is updated via regret matching:

$$\pi^{t+1}(I, a) \propto \max(R^t(I, a), 0) \tag{40}$$

CFR guarantees convergence to Nash equilibria in two-player zero-sum games at a rate of $O(1/\sqrt{T})$, with time-averaged strategies converging to the set of correlated equilibria

in general-sum games.

Deep CFR. Brown et al. (2019a) extended CFR to large-scale games using function approximation. Deep CFR approximates the advantages (counterfactual regrets) at each information set using neural networks:

$$A^t(I, a) \approx D_\theta^t(I, a) \tag{41}$$

where D_θ^t is a deep neural network trained on sampled advantages. The training objective is given by:

$$L(\theta) = \mathbb{E}_{(I,a,v)\sim\mathcal{B}}[(D_\theta(I, a) - v)^2] \tag{42}$$

where $\mathcal{B}$ is a buffer of advantage samples. The average strategy is similarly approximated with a separate neural network trained to minimize:

$$L(\phi) = \mathbb{E}_{(I,\pi)\sim\mathcal{S}}[D_{\text{KL}}(\pi||\bar{\pi}_\phi(I))] \tag{43}$$

where $\mathcal{S}$ is a buffer of strategy samples. Deep CFR has demonstrated remarkable success in large games like no-limit Texas hold'em poker, achieving superhuman performance (Brown et al., 2019b). The combination of regret minimization theory with modern deep learning offers both theoretical guarantees and practical scalability.

These algorithms have achieved significant successes in challenging domains. Deep CFR and variants have demonstrated superhuman performance in poker (Moravčík et al., 2017). Policy gradient methods have proven effective in concave Cournot games (Letcher et al., 2019). NFSP has shown strong performance across imperfect-information games, though generally outperformed by CFR-based methods in poker variants (Moravčík et al., 2017). Regret minimization techniques have been successfully applied beyond poker to games including Starcraft and Stratego (Perolat et al., 2022).

7 Economic Models for Reinforcement Learning

Multi-armed bandit (MAB) problems represent a class of state-less reinforcement learning tasks where an agent repeatedly selects among different actions (arms) to maximize cumulative reward. Unlike regular reinforcement learning which are run in simulation, ban-

dits are typically deployed in real-time which implies that sample efficiency and optimal exploration is critical. Integrating economic structure can improve bandit performance.

The central concept to evaluate bandits is *regret*, which measures the difference between the expected reward under an optimal strategy and the expected reward of the algorithm:

$$R(T) = T \cdot \max_i \mu_i - \mathbb{E}\left[\sum_{t=1}^{T} r_t\right] \tag{44}$$

where μ_i is the expected reward of arm i, and r_t is the observed reward at time t. Lower regret indicates better performance, with $O(\sqrt{T})$ representing a standard baseline for general bandit problems, and improvements to $O(\log T)$ indicating exponentially faster learning.

7.1 Dynamic Pricing with Demand Learning

Consider a seller facing sequential customers with unknown demand. Let $\mathcal{P} = \{p_1, \ldots, p_K\}$ be a discrete price set. At time t, setting price p yields a sale with probability $\theta(p)$ and revenue $R_t = p \cdot \mathbb{1}\{\text{sale}\}$. Standard bandits would treat each price as independent, requiring $\Omega(K)$ exploration rounds. Microeconomic theory contributes the structural assumption that $\theta(p)$ decreases monotonically with p (consumers buy when valuation exceeds price). Under regularity conditions (monotone hazard rate), expected revenue $p\theta(p)$ is unimodal in p. Misra et al. (2019) exploit this structure with an upper-confidence bound algorithm computing an index for each price:

$$I_p(t) = \hat{\mu}_p(t) + \sqrt{\frac{2 \ln t}{N_p(t)}} \tag{45}$$

where $\hat{\mu}_p(t)$ is the average revenue for price p and $N_p(t)$ the number of times p was tried. This achieves polylogarithmic regret compared to $\Omega(\sqrt{T})$ for standard bandits, representing an exponential improvement in learning efficiency. Their field experiments show 43% higher profits compared to static pricing during a month-long deployment.

For multi-product settings, Mueller et al. (2019) consider demand:

$$\mathbf{q}_t = \mathbf{c}_t - \mathbf{B}_t \mathbf{p}_t + \boldsymbol{\varepsilon}_t \tag{46}$$

where $\mathbf{q}_t \in \mathbb{R}^N$ is demand, $\mathbf{B}_t \in \mathbb{R}^{N \times N}$ is the price-sensitivity matrix, and $\mathbf{c}_t$ represents baseline demand. The key economic insight is imposing low-rank structure on $\mathbf{B}_t$, reflecting that cross-price elasticities are driven by a small number of latent factors. Their OPOL algorithm projects high-dimensional price vectors into a learned latent subspace, achieving regret of $O(T^{3/4}\sqrt{d})$ that scales with latent dimension $d \ll N$ rather than product count N. This enables efficient learning even with thousands of products. Xu et al. (2021) demonstrate that economic structure enables logarithmic regret even in adversarial settings. For contextual pricing where valuation $v_t = \boldsymbol{\theta}^\top \mathbf{x}_t$ follows a linear model with feature vector $\mathbf{x}_t$, they achieve optimal $O(d \log T)$ regret—an exponential improvement over standard $\Omega(\sqrt{T})$ bounds for adversarial contexts.

Goyal et al. (2022) address multi-product pricing under the multinomial logit (MNL) choice model, where a consumer's utility for product i is:

$$U_i(\mathbf{p}) = \alpha_i - \beta_i p_i + \epsilon_i \tag{47}$$

and purchase probabilities follow:

$$\Pr(i|\mathbf{p}) = \frac{e^{U_i(\mathbf{p})}}{\sum_j e^{U_j(\mathbf{p})}} \tag{48}$$

By incorporating this MNL structure into an Online Newton Step method with random price perturbations, they achieve $O(d\sqrt{T}\log T)$ regret despite adversarial context arrivals.

7.2 Bid Optimization in Auctions

In a second-price auction with reserve r, given bid profile $\mathbf{b} = (b_{(1)}, b_{(2)}, \ldots)$ (ordered decreasingly), revenue is:

$$\text{Revenue}(r, \mathbf{b}) = b_{(2)} \mathbb{1}\{b_{(2)} > r\} + r \mathbb{1}\{b_{(1)} \geq r > b_{(2)}\} \tag{49}$$

Myerson's auction theory establishes that under regularity conditions, expected revenue $R(r)$ is unimodal in r. Cesa-Bianchi et al. (2015) leverage this in a unimodal bandit algorithm maintaining two UCB indices bracketing the current best reserve, achieving $O(\log T)$ regret in favorable cases—a significant improvement over the $O(\sqrt{T})$ regret of

standard bandits. Akcay et al. (2022) address the partial feedback challenge where losing bids are unobserved. Their algorithm incorporates auction rules into inference: if a sale occurs at price r, exactly one bid exceeded r, informing future exploration. These auction-specific inference rules convert censored observations into side information, accelerating learning in both stationary and non-stationary environments.

7.3 Budget Constraints

Many economic decisions involve resource constraints. Consider an advertiser bidding in repeated auctions with budget B. Let $X_t(a)$ and $C_t(a)$ denote the reward (e.g., clicks) and cost when taking action a at time t. The objective is to maximize $\sum_{t=1}^{T} X_t(a_t)$ subject to $\sum_{t=1}^{T} C_t(a_t) \leq B$. Badanidiyuru et al. (2013) frame this as Bandits with Knapsacks (BwK) and introduce a primal-dual approach with dual price λ_t for the budget constraint. Given estimates of arm reward $\hat{\mu}_i$ and cost $\hat{c}_i$, the algorithm selects the arm maximizing $\hat{\mu}_i - \lambda_t \hat{c}_i$, capturing the economic principle of marginal value versus marginal cost. This achieves near-optimal $\tilde{O}(\sqrt{T})$ regret while respecting budget constraints.[4]

Flajolet and Jaillet (2017) extend this to contextual bidding for online advertising where click-through and cost distributions depend on observable features $\mathbf{x}_t$. Their UCB-based algorithm combines linear payoff estimation with stochastic binary search, achieving regret $\tilde{O}(d\sqrt{T})$ when the budget scales linearly with T. For combinatorial settings with multiple simultaneous auctions, Sankararaman and Slivkins (2018) exploit economic structure by decoupling auctions but coupling their budgets through a shared Lagrange multiplier. This yields regret that grows polynomially rather than exponentially in the number of auctions, making previously intractable problems computationally feasible. Nuara et al. (2018) apply similar principles to joint bid and budget optimization across multiple advertising channels. Using Gaussian Process regression to model each subcampaign's click-through function and dynamic programming to enforce budget constraints, they achieved the same conversion volume as human experts with half the cost per acquisition in a two-month field experiment.

[4]The notation $\tilde{O}$ suppresses logarithmic factors, i.e., $\tilde{O}(f(T)) = O(f(T) \cdot \text{polylog}(T))$.

7.4 Strategic Interactions

Economic theory enhances multi-agent reinforcement learning in competitive markets. Guo et al. (2023) study dynamic pricing between firms where consumers follow a logit model with reference price effects:

$$U_t(i) = \alpha_i - \beta p_{i,t} + \gamma(r_t - p_{i,t}) + \epsilon_{i,t} \tag{50}$$

where r_t is the reference price evolving according to:

$$r_{t+1} = (1-\delta)r_t + \delta \frac{1}{n} \sum_i p_{i,t} \tag{51}$$

By incorporating this behavioral economic insight into online projected gradient ascent, they prove $O(1/t)$ convergence to Nash equilibrium, substantially faster than model-free approaches that lack economic structure.

7.5 Causal Inference

Economic data often contains unobserved confounders that affect both actions and outcomes. Several recent works incorporate econometric identification strategies into reinforcement learning to address these challenges.

Liao et al. (2024) address confounding in offline RL by importing the econometric idea of instrumental variables (IV). They consider a setting where an agent learns from observational data in a Markov decision process with unobserved confounders affecting both actions and rewards. The paper defines valid IVs in the RL context as variables that influence state transitions only through their effect on the action. Using IVs, the authors derive conditional moment restrictions that identify the true transition dynamics despite unobserved confounders. Their IV-aided value iteration (IVVI) algorithm solves a primal-dual optimization problem:

$$\hat{\boldsymbol{\theta}} = \arg\min_{\boldsymbol{\theta}} \max_{\boldsymbol{\lambda}} \mathbb{E}_{(s,a,s',z)} \left[\boldsymbol{\lambda}(s,z)^\top (s' - f_{\boldsymbol{\theta}}(s,a)) \right] \tag{52}$$

This represents the first offline RL algorithm to leverage instruments for consistent value learning, combining causal inference tools with reinforcement learning.

Bennett and Kallus (2023) propose a "proximal reinforcement learning" (PRL) frame-

work that adapts proximal causal inference to RL. They focus on off-policy evaluation in partially observed MDPs, where hidden state variables confound the reward dynamics. The key insight is to assume the existence of certain bridge functions that link observed data to the target policy value. Under appropriate conditions, these functions allow identification of the true policy value from confounded data:

$$V(\pi) = \mathbb{E}\left[h(X, W, A) \mid Z = z\right] \tag{53}$$

where h is the bridge function, X represents observed states, W unobserved confounders, A the action, and Z a proxy variable. The authors construct estimators using these bridge functions and prove they attain semiparametric efficiency, demonstrating how incorporating structural assumptions leads to more accurate policy evaluation.

Wang et al. (2023) introduce "Super RL," a paradigm for offline reinforcement learning that incorporates expert actions as an additional input to overcome unmeasured confounding. In many economic settings (e.g., medical treatment, policy interventions), historical data come from human experts whose decisions may contain information about unobserved factors. The super policy maps both the observed state and the expert action to a new action:

$$\pi_{\text{super}}(s, a_{\text{expert}}) \to a \tag{54}$$

The authors establish identification conditions under which the super policy is guaranteed to strictly dominate both standard RL policies and expert policies in terms of performance. Their method effectively embeds the rationality of human experts and their private information into the RL training process, greatly improving sample efficiency and policy quality.

7.6 Discussion of Algorithmic Improvements

The integration of economic structure into reinforcement learning produces significant theoretical improvements across multiple domains. When demand curves satisfy regularity conditions, regret bounds improve from $O(\sqrt{T})$ to $O(\log T)$ or $O(\text{poly}(\log T))$. For multi-product settings, low-rank structure reduces dependence from product count N to intrinsic dimension d. These theoretical gains translate to practical performance im-

provements. Structure-aware algorithms learn optimal policies with orders of magnitude fewer samples than model-free reinforcement learning. By leveraging domain knowledge about utility maximization, budget constraints, revealed prefernece, market equilibria, rational expectations or behavioural biases (e.g. representativeness or conservatism) we can improve the performance of reinforcement learning.

8 Real World Applications and Deployments

In this section, we cover the most enduring economic applications of SE and RL. We note that SE has a long history in influencing public policy, while RL has only recently begun to see deployment in business and economic applications.

8.1 Economic Applications of Structural Econometrics

Structural econometric models have influenced policy decisions across various economic domains. In macroeconomic policy, central banks employ structural models as core analytical tools. The Federal Reserve's FRB/US model and the European Central Bank's New Area-Wide Model provide frameworks for monetary policy analysis through counterfactual simulations (Fischer, 2017; Coenen et al., 2018). As former Federal Reserve Vice Chair Stanley Fischer observed, these models serve as "key tools" for scenario analysis at major central banks (Fischer, 2017). Similarly, fiscal policy has incorporated structural approaches through dynamic scoring of tax legislation. The analysis of the 2017 Tax Cuts and Jobs Act employed overlapping-generations models to quantify macroeconomic feedback effects, projecting modest GDP increases and additional tax revenue (U.S. Congress, 2017).

Antitrust enforcement has been transformed by structural merger simulation, providing quantitative evidence for regulatory decisions. In U.S. v. H&R Block (2011), the court explicitly referenced the Justice Department's structural model forecasts in blocking the acquisition of TaxACT (Sonsini, 2012). The European Commission has similarly utilized structural models in telecommunications merger reviews, including the Hutchison/Telefonica UK case, where simulations projected potential price increases that influenced regulatory outcomes (Practical Law Competition, 2020; Botts, 2020).

Market design represents a direct application of structural economics, with theoret-

ical models creating new institutions. The FCC's spectrum auctions exemplify this approach, with auction mechanisms designed to efficiently reallocate radio spectrum (Federal Communications Commission, 2020). Educational assignment systems have implemented matching algorithms based on the Gale-Shapley deferred acceptance mechanism, substantially reducing the number of unmatched students when adopted in New York City schools. Similar mechanisms have been employed in healthcare for kidney exchange programs, increasing transplantation rates through multi-patient matching chains.

Social policy decisions often rely on structural models to predict behavioral responses to program changes. The Affordable Care Act's design was informed by microsimulation models projecting coverage expansion and budget impacts (Rosenbaum, 2011). Labor market policies, including minimum wage regulations, have been evaluated using structural models that estimate employment effects and income distribution changes (NPR, 2025). These quantitative predictions provide policymakers with evidence-based projections of reform consequences, though the accuracy of such forecasts remains subject to ongoing empirical validation.

8.2 Economic Applications of Reinforcement Learning

Reinforcement learning techniques are beginning to be applied to economic decisions at scale, replacing rule-based or simpler learning methods.

In transportation, DiDi implemented multi-agent RL for ride-hailing dispatch optimization. Their system, deployed across multiple Chinese cities, achieved 0.5-2% improvements in key metrics including driver income and order completion rates compared to previous production baselines (Wang et al., 2019b). At DiDi's scale, these seemingly modest gains translate to substantial efficiency improvements.

Microsoft applied RL to advertising content personalization, notably for Xbox homepage promotions. Their system dynamically selected which promotional content to display to each user based on contextual information. In live A/B testing, the RL-driven approach yielded a 60% higher click-through rate compared to baseline policies (Ie et al., 2022), demonstrating the effectiveness of sequential decision-making in user engagement optimization.

In finance, JPMorgan Chase developed LOXM, an RL-driven algorithm for executing large equity orders. Trained on billions of historical trades, the system optimizes exe-

cution timing and sizing to minimize market impact while maximizing price efficiency. When deployed on trading desks, LOXM achieved approximately 15% better execution efficiency compared to traditional strategies (Patel, 2018), representing one of the first production applications of deep RL in institutional finance.

For e-commerce, Alibaba leveraged RL for dynamic pricing during major shopping events. During their 2018 "Double 11" Shopping Festival, which generated $30.8 billion in 24 hours, Alibaba deployed RL agents that forecasted product demand and automatically adjusted prices in real-time (Wang et al., 2019a). This enabled optimal discount strategies and inventory allocation, contributing to a 27% year-over-year growth in sales.

Marketing applications include attempts to optimize customer interactions. Liu et al. (2022) compared model-free RL policies for coupon targeting against conventional approaches in controlled experiments, reporting modest improvements in customer response. Misra et al. (2019)'s successfully demonstrate the value of incorporating incorporating microeconomic choice theory into scalable dynamic price experimentation.

Google's application of RL to data center cooling demonstrates impact in operational efficiency. By using deep RL to autonomously control cooling equipment in production data centers, Google reduced cooling energy consumption by up to 40% (Gao, 2018). The system continually adjusts fans, chillers, and other equipment to maintain safe temperatures while minimizing energy usage, yielding a 15% reduction in overall Power Usage Effectiveness (PUE).

9 Conclusions and Discussion

This paper has explored the methodological synergies between structural econometrics and reinforcement learning. We demonstrated that reinforcement learning algorithms can be viewed as generalizations of dynamic programming methods central to structural economics. RL can be used to solve static and dynamic games, static and dynamic decision making problems.

The cross-fertilization between these disciplines offers substantial benefits in both directions. Structural econometrics gains computational tools to solve, high-dimensional models previously considered intractable, while reinforcement learning benefits from economic theory's structural assumptions. One of the reasons why AlphaZero did so well

was because it had a perfect model.

An approximate and reasonably accurate model can significantly reduce the search space for learning. The success of RL in games and robotics is attributable to their existing perfect models of games and near perfect physics simulators. In economics and social sciences, we cannot expect to build perfect simulators but we can capture and exploit some empirical regularities (e.g. demand, budget constraints, loss aversion) and incorporate them to improve real-time learning algorithms.

Methodological challenges will persist. RL is brittle, requires extensive hyperparameter tuning, lacks standardized implementations, is computationally demanding, and convergence is not guaranteed in many settings. Nevertheless, it offers a potentially powerful toolkit for dynamic and strategic decision-making.

References

Pieter Abbeel and Andrew Y. Ng. Apprenticeship learning via inverse reinforcement learning. *Proceedings of the International Conference on Machine Learning (ICML)*, 2004.

Daniel A. Ackerberg, Kevin Caves, and Garth Frazer. Identification properties of recent production function estimators. *Econometrica*, 83(6):2411–2451, 2015.

S. Adusumilli, M. Eckardt, and G. Tate. Estimation of dynamic discrete choice models with differentiable temporal-difference learning. *arXiv preprint arXiv:2209.15174*, 2022.

Rishabh Agarwal, Dale Schuurmans, and Mohammad Norouzi. An optimistic perspective on offline reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2020.

Victor Aguirregabiria and Pedro Mira. Sequential estimation of dynamic discrete games. *Econometrica*, 75(1):1–53, 2007.

Alp Akcay, Onur Atan, Muhammed O. Sayin, and Atilla Eryilmaz. Online learning algorithms for auction reserve price optimization with censored feedback. *Operations Research*, 2022.

John Asker, Chaim Fershtman, Jihye Jeon, and Ariel Pakes. A computational framework for analyzing dynamic auctions: The market impact of information sharing. *The RAND Journal of Economics*, 51(3):805–839, 2020.

Tohid Atashbar and Shuping Shi. Solving macroeconomic models with deep reinforcement learning. *Journal of Economic Dynamics and Control*, 2023.

Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47(2):235–256, 2002.

Ashwinkumar Badanidiyuru, Robert Kleinberg, and Aleksandrs Slivkins. Bandits with knapsacks. *IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 207–216, 2013.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Benjamin Lanham, Carrick Tian, Pranav Baljekar, Shauna Huang, Sheer El Gonzalez, Paul Christiano, Jan Leike, and Ryan Krueger. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

Patrick Bajari, C. Lanier Benkard, and Jonathan Levin. Estimating dynamic models of imperfect competition. *Econometrica*, 75(5):1331–1370, 2007.

Andrew G. Barto, Richard S. Sutton, and Charles W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(5):834–846, 1983.

Richard Bellman. *Dynamic Programming*. Princeton University Press, 1957.

Daniel T. Bennett and Nathan Kallus. Proximal reinforcement learning: A bridge between causal inference and sequential decision making. *arXiv preprint arXiv:2306.12351*, 2023.

Steven Berry, James Levinsohn, and Ariel Pakes. Automobile prices in market equilibrium. *Econometrica*, 63(4):841–890, 1995.

Dimitri P. Bertsekas. *Dynamic Programming and Optimal Control*. Athena Scientific, 1996.

Dimitri P. Bertsekas and John N. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Scientific, Belmont, MA, 1996.

Nicolás Bogota, Xiaocheng Huang, and Tony Jebara. Prediction of refugee migration patterns using maximum entropy inverse reinforcement learning. *NIPS Workshop on Machine Learning for Social Good*, 2015.

Tilman Börgers and Rajiv Sarin. Learning through reinforcement and replicator dynamics. *Journal of Economic Theory*, 77(1):1–14, 1997.

Baker Botts. The hutchison judgment, December 2020. URL https://www.bakerbotts.com/~/media/Files/Thought-Leadership/Publications/2020/December/The-Hutchinson.pdf.

Abdeslam Boularias, Jens Kober, and Jan Peters. Relative entropy inverse reinforcement learning. *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, 2011.

Michael Bowling and Manuela Veloso. Multiagent learning using a variable learning rate. *Artificial Intelligence*, 136(2):215–250, 2002.

Ronen I. Brafman and Moshe Tennenholtz. R-max – a general polynomial time algorithm for near-optimal reinforcement learning. *Journal of Machine Learning Research*, 3:213–231, 2002.

Gianluca Brero, Alon Eden, Matthias Gerstgrasser, David C. Parkes, and Duncan Rheingans-Yoo. Reinforcement learning of sequential price mechanisms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 13662–13670, 2021.

Yoav Bronner, Amir Fishman, Ya'acov Ritov, and Ilan Shimshoni. Detecting anomalous ride-hailing driver routes using inverse reinforcement learning. *Transportation Research Part C: Emerging Technologies*, 146:103982, 2023.

George W. Brown. Iterative solution of games by fictitious play, 1951.

Noam Brown, Adam Lerer, Sam Gross, and Tuomas Sandholm. Deep counterfactual regret minimization. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97, pages 793–802. PMLR, 2019a.

Noam Brown, Adam Lerer, Sam Gross, and Tuomas Sandholm. Deep counterfactual regret minimization. In *Proceedings of the 36th International Conference on Machine Learning (ICML)*, volume 97, pages 793–802. PMLR, 2019b.

Haoyang Cao, Samuel N. Cohen, and Lukasz Szpruch. Identifiability in inverse reinforcement learning. In *Advances in Neural Information Processing Systems 34 (NeurIPS 2021)*, 2021.

Nicolo Cesa-Bianchi, Claudio Gentile, and Yishay Mansour. Regret minimization for reserve prices in second-price auctions. *IEEE Transactions on Information Theory*, 61 (1):549–564, 2015.

David C. Chan, Matthew Gentzkow, and Chuan Yu. Selection with variation in diagnostic skill: Evidence from radiologists. *The Quarterly Journal of Economics*, 137(2):729–783, 2019.

Federico Ciliberto and Elie Tamer. Market structure and multiple equilibria in airline markets. *Econometrica*, 77(6):1791–1828, 2009.

Günter Coenen, Peter Karadi, Sebastian Schmidt, and Anders Warne. The new area-wide model ii: an extended version of the ecb's micro-founded model for forecasting and policy analysis with a financial sector. Technical Report 2200, European Central Bank, November 2018. URL https://www.ecb.europa.eu/pub/pdf/scpwps/ecb.wp2200.en.pdf.

Bryan S. Covarrubias, Alexander Zentefis, and Jose M. Abiseid. Collusion with deep reinforcement learning. Technical Report w30283, National Bureau of Economic Research, July 2022.

John G. Cross. A stochastic learning model of economic behavior. *The Quarterly Journal of Economics*, 87(2):239–266, 1973.

Michael D. Curry, Carsenate Banerjee, Yan Li, Paul Evans, Iurii Svitov, and Stephen Zhang. Finding equilibrium in heterogeneous agent models with deep reinforcement learning. In *AI for Agent-Based Modelling (AI4ABM) Workshop at AAMAS*, 2022.

Constantinos Daskalakis, Paul W. Goldberg, and Christos H. Papadimitriou. The complexity of computing a nash equilibrium. *SIAM Journal on Computing*, 39(1):195–259, 2009.

Jean-Pierre Dubé, Jeremy T. Fox, and Che-Lin Su. Improving the numerical performance of static and dynamic aggregate discrete choice random coefficients demand estimation. *Econometrica*, 80(5):2231–2267, 2012.

Stefano Ermon, Yexiang Xue, Russell Toth, Bistra Dilkina, Richard Bernstein, Theodoros Damoulas, Patrick Clark, Steve DeGloria, Andrew Mude, Christopher Barrett, and Carla P. Gomes. Learning large-scale dynamic discrete choice models of spatio-temporal preferences with application to migratory pastoralism in east africa. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2015.

Federal Communications Commission. Auction 97: Advanced wireless services (aws-3), November 2020. URL https://www.fcc.gov/auction/97.

Jesús Fernández-Villaverde, Galo Nuño, and Jesse Perla. Solving high-dimensional dynamic programming problems using deep learning. *Journal of Econometrics*, 2024.

Chaim Fershtman and Ariel Pakes. Dynamic games with asymmetric information: A framework for empirical work. *The Quarterly Journal of Economics*, 127(4):1611–1661, 2012.

Chelsea Finn, Sergey Levine, and Pieter Abbeel. Guided cost learning: Deep inverse optimal control via policy optimization. *Proceedings of the International Conference on Machine Learning (ICML)*, 2016.

Stanley Fischer. I'd rather have bob solow than an econometric model, but... Speech at the Warwick Economics Summit, Coventry, United Kingdom, February 2017. URL https://www.bis.org/review/r170214a.htm.

Arthur Flajolet and Patrick Jaillet. Real-time bidding with side information. *Advances in Neural Information Processing Systems*, 30, 2017.

Justin Fu, Katie Luo, and Sergey Levine. Learning robust rewards with adversarial inverse reinforcement learning. *International Conference on Learning Representations (ICLR)*, 2018.

Jim Gao. Data center cooling using model-predictive control. *Advances in Neural Information Processing Systems*, 31, 2018.

J. Gijsbrechts, R. N. Boute, J. A. Van Mieghem, and D. J. Zhang. Can deep reinforcement learning improve inventory management? performance on lost sales, dual-sourcing, and multi-echelon problems. *Manufacturing & Service Operations Management*, 24(3), 2022.

Vikas Goyal, Negin Jain, and Aleksandrs Slivkins. Online learning for multi-product dynamic pricing with multinomial logit choice model. *Operations Research*, 2022.

Lukas Graf, Patrick Hummel, and Martin Bichler. Deep reinforcement learning for auctions: Evaluating bidding strategies effectiveness and convergence. In *Proceedings of the 17th International Conference on Agents and Artificial Intelligence (ICAART 2025)*. SCITEPRESS - Science and Technology Publications, 2025. Forthcoming.

Yanjun Guo, Yash Kanoria, and David Simchi-Levi. Multi-agent reinforcement learning in dynamic pricing games. *Management Science*, 2023.

Trygve Haavelmo. The probability approach in econometrics. *Econometrica*, 12:1–115, 1944.

James J. Heckman. Sample selection bias as a specification error. *Econometrica*, 47(1): 153–161, 1979.

Johannes Heinrich and David Silver. Deep reinforcement learning from self-play in imperfect-information games. In *NIPS Deep Reinforcement Learning Workshop*, 2016.

Natascha Hinterlang and Tobias Taenzer. Optimal monetary policy using reinforcement learning. *Journal of Monetary Economics*, 2024.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. *Advances in Neural Information Processing Systems*, 2016.

Brett Hollenbeck. Horizontal mergers and innovation in concentrated industries. *Quantitative Marketing and Economics*, 2019.

William C. Hood and Tjalling C. Koopmans. *Studies in Econometric Method*. John Wiley & Sons, 1953.

V. Joseph Hotz and Robert A. Miller. Conditional choice probabilities and the estimation of dynamic models. *The Review of Economic Studies*, 60(3):497–529, 1993.

Ronald A. Howard. *Dynamic Programming and Markov Processes*. The Technology Press of M.I.T. and John Wiley and Sons, New York, NY, 1960.

Junling Hu and Michael P Wellman. Nash q-learning for general-sum stochastic games. *Journal of Machine Learning Research*, 4(Nov):1039–1069, 2003.

Yingyao Hu and Zhangyi Yang. Structural estimation with policy gradient methods. *Working Paper*, 2025.

Eugene Ie, Vihan Jain, Jing Wang, Sanmit Narvekar, Ritesh Agarwal, Rui Wu, Heng-Tze Cheng, Morgane Lustman, Vince Gatto, Paul Covington, et al. Reinforcement learning for xbox live content recommendations. In *Proceedings of the 16th ACM Conference on Recommender Systems*, pages 1–9, 2022.

Mitsuru Igami. Artificial intelligence as structural estimation: Deep blue, bonanza, and alphago. *The Econometrics Journal*, 23(3):S1–S24, 2020.

Sham M. Kakade. A natural policy gradient. *Advances in Neural Information Processing Systems*, 14, 2001.

Kuno Kim, Kirankumar Shiragur, Shivam Garg, and Stefano Ermon. Reward identification in inverse reinforcement learning. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139, pages 5496–5505. PMLR, 2021.

Ilya Kostrikov, Kumar Krishna Agrawal, Debidatta Dwibedi, Sergey Levine, and Jonathan Tompson. Discriminator-actor-critic: Addressing sample inefficiency and reward bias in adversarial imitation learning. *International Conference on Learning Representations (ICLR)*, 2018.

Antony Letcher, David Balduzzi, Ian Gemp, Jakob N Foerster, Wojciech M Czarnecki, Karl Tuyls, and Thore Graepel. Differentiable game mechanics. In *International Conference on Learning Representations (ICLR)*, 2019.

Ziyan Liao, Ziqi An, Xiaoxiao Wu, Yang Liu, and Min Yin. Offline reinforcement learning with instrumental variables in confounded markov decision processes. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Michael L Littman. Markov games as a framework for multi-agent reinforcement learning. In *Machine Learning Proceedings 1994*, pages 157–163. Morgan Kaufmann, 1994.

Xiao Liu, Zhengling Qin, Xiaoying Gao, and Junming Huang. Dynamic coupon targeting using batch deep reinforcement learning: An application to livestream shopping. *Management Science*, 2022.

Nikolay Lomys and Luca Magnolfi. Estimation of games under no regret: Structural econometrics for ai. Working Paper NET Institute Working Paper No. 24-05, Social Science Research Network (SSRN), 2024. Available at SSRN: https://ssrn.com/abstract=4717195 or http://dx.doi.org/10.2139/ssrn.4717195.

Tien Mai, Mogens Fosgerau, and Emma Frejinger. A nested recursive logit model for route choice analysis. *Transportation Research Part B: Methodological*, 75:100–112, 2015.

Lilia Maliar, Serguei Maliar, and Pablo Winant. Deep learning for solving dynamic economic models. *Journal of Monetary Economics*, 122:76–101, 2021.

Eric Mazumdar and et al. On gradient-based learning in continuous games. *arXiv preprint arXiv:1906.01217*, 2019.

Daniel McFadden. Conditional logit analysis of qualitative choice behavior. *Frontiers in Econometrics*, pages 105–142, 1974.

Robert A. Miller. Job matching and occupational choice. *Journal of Political Economy*, 92(6):1086–1120, 1984.

Marvin Minsky. *Neural Nets and the Brain Model Problem.* PhD thesis, Princeton University, 1954.

Kanishka Misra, Eric M. Schwartz, and Jacob Abernethy. Dynamic online pricing with incomplete information using multiarmed bandit experiments. *Marketing Science*, 38 (2):226–252, 2019.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518:529–533, 2015.

Dov Monderer and Lloyd S. Shapley. Potential games. *Games and Economic Behavior*, 14(1):124–143, 1996.

Matěj Moravčík, Martin Schmid, Neil Burch, Viliam Lisý, Dustin Morrill, Nolan Bard, Trevor Davis, Kevin Waugh, Michael Johanson, and Michael Bowling. Deepstack: Expert-level artificial intelligence in no-limit poker. *Science*, 356(6337):508–513, 2017.

Jonas Mueller, Vasilis Syrgkanis, and Matt Taddy. Low-rank bandit methods for high-dimensional dynamic pricing. *Advances in Neural Information Processing Systems*, 32, 2019.

Andrew Y. Ng and Stuart J. Russell. Algorithms for inverse reinforcement learning. *Proceedings of the International Conference on Machine Learning (ICML)*, 2000.

Y. Nomura et al. Deep reinforcement learning for dynamic pricing and ordering policies in perishable inventory management. *Applied Sciences*, 15(5):2421, 2025.

NPR. 21 states are getting minimum wage bumps in 2025. *NPR*, January 2025. URL https://www.npr.org/2025/01/01/nx-s1-5244050/states-minimum-wage-increase-2025.

Alessandro Nuara, Francesco Trovo, Nicola Gatti, and Marcello Restelli. A combinatorial-bandit algorithm for the online joint bid/budget optimization of pay-per-click advertising campaigns. *AAAI Conference on Artificial Intelligence*, 32(1), 2018.

Jackson B. O'Briant. Inverse reinforcement learning for structural models. Job Market Paper, 2024.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. *arXiv preprint arXiv:2203.02155*, 2022.

Marko Patel. J.p. morgan's massive guide to machine learning and big data in the financial services industry. *Business Insider*, 2018.

Jean Perolat, Remi Munos, Jean-Baptiste Lespiau, Mathis Lauriere, Rémi Begu, Bilal Piot, Viliam Lisy, Karl Tuyls, Nicolò De Lazzer, Laurent Orseau, et al. Mastering the game of stratego with model-free multiagent reinforcement learning. *Science*, 378 (6622):890–896, 2022.

Martin Pesendorfer and Philipp Schmidt-Dengler. Asymptotic least squares estimators for dynamic games. *The Review of Economic Studies*, 75(3):901–928, 2008.

Amil Petrin. Quantifying the benefits of new products: The case of the minivan. *Journal of Political Economy*, 110(4):705–729, 2002.

Practical Law Competition. Commission decision to prohibit hutchison 3g uk/telefonica uk merger annulled (general court), May 2020. URL https://content.next.westlaw.com/practical-law/document/I46642dd8a0a111eabea3f0dc9fb69570/Commission-decision-to-prohibit-Hutchison-3G-UK-Telefonica-UK-merger-annulled-Gener

Deepak Ramachandran and Eyal Amir. Bayesian inverse reinforcement learning. *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*, 2007.

Arunselvan Ramaswamy. Analyzing approximate value iteration algorithms. *arXiv preprint arXiv:1709.04673*, 2021. v5, revised 30 May 2021.

Nathan D. Ratliff, J. Andrew Bagnell, and Martin A. Zinkevich. Maximum margin planning. *Proceedings of the International Conference on Machine Learning (ICML)*, 2006.

Sai Srivatsa Ravindranath, Zhe Feng, Di Wang, Manzil Zaheer, Aranyak Mehta, and David C. Parkes. Deep reinforcement learning for sequential combinatorial auctions. Submitted to ICLR 2025, 2024. Available at https://openreview.net/forum?id=SVd9Ffcdp8.

Julia Robinson. An iterative method of solving a game. *Annals of Mathematics*, pages 296–301, 1951.

Hadrien Rolland, Matteo Pirotta, Yannis Flet-Berliac, and Philippe Preux. Identifiability of rewards in inverse reinforcement learning. *arXiv preprint arXiv:2202.09529*, 2022a.

Hadrien Rolland, Matteo Pirotta, Yannis Flet-Berliac, and Philippe Preux. Identifiability of rewards in inverse reinforcement learning. *arXiv preprint arXiv:2202.09529*, 2022b.

Sara Rosenbaum. The patient protection and affordable care act. *PubMed Central*, 126(1):130–135, January 2011. URL https://pmc.ncbi.nlm.nih.gov/articles/PMC3001814/.

John Rust. Optimal replacement of gmc bus engines: An empirical model of harold zurcher. *Econometrica*, 55(5):999–1033, 1987.

M. Sabri et al. Reinforcement learning and stochastic dynamic programming for jointly scheduling jobs and preventive maintenance on a single machine to minimise earliness-tardiness. *Production Research*, 62(3):705–719, 2024.

Arthur L. Samuel. Some studies in machine learning using the game of checkers. *IBM Journal of Research and Development*, 3(3):210–229, 1959.

Karthik Abinav Sankararaman and Aleksandrs Slivkins. Combinatorial semi-bandits with knapsacks. *International Conference on Artificial Intelligence and Statistics*, pages 1760–1770, 2018.

John Schulman, Sergey Levine, Pieter Abbeel, Michael Jordan, and Philipp Moritz. Trust region policy optimization. *Proceedings of the 32nd International Conference on Machine Learning*, pages 1889–1897, 2015.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Claude E. Shannon. Programming a computer for playing chess. *Philosophical Magazine*, 41(314), 1950.

Lloyd S. Shapley. Some topics in two-person games. In M. Dresher, L. S. Shapley, and A. W. Tucker, editors, *Advances in Game Theory*, pages 1–28. Princeton University Press, 1964.

Mohit Sharma, Arjun Sharma, Nicholas Rhinehart, and Kris M. Kitani. Directed-info gail: Learning hierarchical policies from unsegmented demonstrations using directed information. *International Conference on Learning Representations (ICLR)*, 2018.

Moussa Shehab, Arunesh Sinha, and Matthew E. Taylor. Identifiability in inverse reinforcement learning. *arXiv preprint arXiv:2401.03608*, 2024.

David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, 2016.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

Satinder Singh, Tommi Jaakkola, Michael L Littman, and Csaba Szepesvári. Convergence results for single-step on-policy reinforcement-learning algorithms. *Machine learning*, 38:287–308, 2000.

Wilson Sonsini. Antitrust agencies action in 2012. *Antitrust Report*, 2012. URL https://www.wsgr.com/a/web/229/pak-0312.pdf.

Richard S. Sutton. Learning to predict by the methods of temporal differences. *Machine Learning*, 3(1):9–44, 1988. doi: 10.1023/A:1022633531479.

Richard S. Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In *Proceedings of the Seventh International Conference on Machine Learning*, pages 216–224. Morgan Kaufmann, 1990.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. MIT Press, Cambridge, MA, 1998.

Richard S. Sutton and Andrew G. Barto. *Reinforcement Learning: An Introduction*. The MIT Press, 2nd edition, 2018.

Richard S. Sutton, David A. McAllester, Satinder P. Singh, and Yishay Mansour. Policy gradient methods for reinforcement learning with function approximation. *Advances in Neural Information Processing Systems*, 12, 1999a.

Richard S. Sutton, Doina Precup, and Satinder Singh. Between mdps and semi-mdps: A framework for temporal abstraction in reinforcement learning. *Artificial Intelligence*, 112(1-2):181–211, 1999b.

Ming Tan. Multi-agent reinforcement learning: Independent versus cooperative agents. Technical Report CMU-CS-93-193, Carnegie Mellon University, School of Computer Science, 1993.

Gerald Tesauro. Td-gammon, a self-teaching backgammon program, achieves master-level play. *Neural Computation*, 6(2):215–219, 1994.

John N. Tsitsiklis. On the convergence of optimistic policy iteration. *Journal of Machine Learning Research*, 3:59–72, 2002.

Karl Tuyls, Ann Nowé, Bernard Manderick, and Katja Verbeeck. An evolutionary game theoretic approach to q-learning in multi-agent systems. In *Proceedings of the Second International Joint Conference on Autonomous Agents and Multiagent Systems*, pages 729–736. ACM, 2003.

U.S. Congress. Tax cuts and jobs act of 2017, 2017. Public Law 115-97.

Yanchao Wang, Gaoyue Qi, and Chen Shi. Super-rl: A general framework for offline reinforcement learning by leveraging supervised learning. *arXiv preprint arXiv:2310.09941*, 2023.

Yongfeng Wang, Kangyi Ouyang, Chunyan Huang, Jian Chen, Yanli Liu, and Ying Wei. Aliexpress dynamic pricing: A learning approach. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 2547–2555, 2019a.

Zhaodong Wang, Zhiwei Qin, Xiaocheng Tang, Jieping Ye, and Hongtu Zhu. Reinforcement learning for order dispatching: A ride-sharing perspective. *arXiv preprint arXiv:1905.11566*, 2019b.

Christopher J. C. H. Watkins and Peter Dayan. Q-learning. *Machine Learning*, 8(3-4): 279–292, 1992. doi: 10.1007/bf00992698.

Gabriel Y. Weintraub, C. Lanier Benkard, and Benjamin Van Roy. Markov perfect industry dynamics with many firms. *Econometrica*, 76(6):1375–1411, 2008.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992a.

Ronald J. Williams. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256, 1992b.

Kenneth I. Wolpin. An estimable dynamic stochastic model of fertility and child mortality. *Journal of Political Economy*, 92(5):852–874, 1984.

Markus Wulfmeier, Peter Ondruska, and Ingmar Posner. Maximum entropy deep inverse reinforcement learning. *arXiv preprint arXiv:1507.04888*, 2015.

Yining Xu, Zhe Wang, and Yuchen Yu. Logarithmic regret in feature-based dynamic pricing. *Advances in Neural Information Processing Systems*, 34, 2021.

Will Yancey, Haotian Bai, and Rema Hanna. Inverse reinforcement learning for labor market dynamics. *Working Paper*, 2022.

Jiawei Zeng, Bin Gu, and Heng Huang. A fully single loop algorithm for bilevel optimization without hessian inverse. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(7):7426–7434, 2022.

Shuai Zhang, Hongkang Li, Meng Wang, Miao Liu, Pin-Yu Chen, Songtao Lu, Sijia Liu, Keerthiram Murugesan, and Subhajit Chaudhury. On the convergence and sample complexity analysis of deep q-networks with ϵ-greedy exploration. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

Brian D. Ziebart, Andrew Maas, J. Andrew Bagnell, and Anind K. Dey. Maximum entropy inverse reinforcement learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 2008.

Martin Zinkevich, Michael Johanson, Michael Bowling, and Carmelo Piccione. Regret minimization in games with incomplete information. In *Advances in Neural Information Processing Systems*, volume 20, 2008.