Sensitivity and Reproducibility: Deep Reinforcement Learning in a Multi-Agent Real Business Cycle Model

Justin Novick - 260965106

Building on 'Analyzing Micro-Founded General Equilibrium Models with Many Agents Using Deep Reinforcement Learning'

Undergraduate Research Project

McGill University

Montreal, Canada

April 30, 2024

1. Abstract

Agent-Based Computational Economics (ACE) has gained popularity over the past two decades. This discipline utilizes simulations to explain environmental outcomes and market phenomena. It accomplishes this by assigning actors rationality at a micro level, rather than imposing equilibrium at specific time steps. This approach is viewed by many researchers as essential for overcoming the Lucas Critique. However, the concept of rationality becomes complex in dynamic, intricate environments.

Reinforcement learning (RL), a subset of machine learning, may solve these complexities. In Analyzing micro-founded general equilibrium models with many agents using deep reinforcement learning, Curry et al. explore a use case of an RL algorithm in an environment reminiscent of the widely recognized macroeconomic Real Business Cycle model. Although the agents in this model adapt well, their behavior and preferences are still somewhat influenced by the predefined utility functions.

This paper tests the findings of Curry et al., and further investigates how altering reward-yielding utility functions affect the overall results in the environment. Experiments were conducted with Constant Relative Risk Aversion (CRRA), Constant Absolute Risk Aversion (CARA), and a hybrid of the two. Furthermore, an analysis is provided as to whether or not agents naturally find a collective equilibrium at the end of a learning curriculum.

2. Introduction

The Real Business Cycle (RBC) model has become one of macroeconomics' most popular tools for explaining market fluctuations, shocks, and frictions. The typical structure of this model includes a labor market, a goods market, and a capital market. To illustrate agents' behavior in these markets, the RBC model uses a representative firm and a representative consumer. Furthermore, certain extensions of this model consider actions of an imagined government, such as taxation. The RBC representation is regarded as a Dynamic General Equilibrium (DGE) model, meaning that analysis imposes that markets eventually clear in response to triggers, and the representative agents are assumed to be rational. The RBC model, of course, has its flaws. The famous Lucas Critique argues that macroeconomic models can lead to misleading policy analysis if they fail to account for changes in agent expectations in response to policy shifts (Lucas, 1976). Following the critique, there has been a push for macroeconomic models to utilize more robust microfoundations (Tilbury, 2023).

The field of Agent-Based Computational Economics (ACE) has arisen in response to the need to capture micro-level activity that, when summed up, contributes to aggregate changes in an economy. In contrast to the use of representative homogenous agents, ACE models typically have several heterogenous agents whose collective behavior is observed. One of many recognizably ambitious projects in this field was the EURACE project, which attempted to model the entire European economy (Dawid et al., 2012). This project fell short of its ambitions, leading to valid criticism of ACE. The primary concern with such projects was that the rule-based agents in the economy were not adapting as expected; their use of a hard-coded priori limited their ability to learn. Thus, such projects failed to capture rational behavior at a micro level (Tesfatsion, 2002). Additionally, there were concerns regarding the computational complexity associated with further advancing such models (Axtell 2000). Luckily, advancements in graphics processing units (GPUs) at the turn of the century have significantly alleviated such concerns. These hardware improvements enable substantially more parallel processing compared to traditional CPUs, enhancing the feasibility of complex Artificial Intelligence (AI) models. A key sub-field of AI addresses the concern of agent adaptability: Reinforcement Learning (RL). This sub-field is particularly involved with optimizing agent policies based on experience interacting within an environment. It has been proposed that integrating RL algorithms into agent decision-making could address the lack of dynamic preferences seen in previous ACE models.

This paper's central experiments investigate a multiagent augmentation of the RBC model in which agents are equipped with RL algorithms. Instead of a representative firm and consumer, the proposed environment has 100 consumers, 10 firms, and 1 government. The use of many agents allows their heterogeneous activity to form the economy's aggregate behavior. Here, equilibrium is not imposed; however, analysis of whether or not agents naturally find one can point towards the predictability of outcomes. This paper first focuses on reproducing the results displayed by Curry et al. Additionally, it is worth noting that the amount of inspiration this environment takes from the traditional RBC model still leaves it reliant on many assumptions. For instance, consumer decisions are still largely influenced by an assumed utility function, which yields rewards for the RL algorithm to train on. This paper further investigates how changing such utility functions affect experimental outcomes. Ultimately, the analysis in this paper investigates whether or not agents learn rational behavior at a micro level in this multi-agent reinforcement learning environment.

3. Literature Review

Reinforcement Learning (RL) is a branch of Machine Learning (ML) in which agents learn to make decisions by taking actions within an environment. Typically, these agents are tasked with exploring a state space and selecting actions that maximize a future stream of rewards, which is discounted back to their present value. The actions taken by an RL agent result in feedback from the environment, which in turn influences future decisions and the refinement of strategies. More technically, RL agents evaluate the state of the environment, often using methods that can approximate the value of different states based on previously observed data. This value function, which adapts with each new observation, is critical for predicting the potential outcomes of various actions. Additionally, RL agents operate under a policy — a set of rules governing action selection. Policies evolve over time and can be represented with probabilistic methods or functional approximations. Through numerous iterations and simulations, the value estimations for environment states converge toward stable values that reflect approximately true rewards. Then, the policies gradually refine toward optimal choice selection. This iterative learning process allows RL agents to adapt and optimize their behavior in complex environments (Sutton 2024). The advancing field of RL has boasted groundbreaking accomplishments such as self-driving cars and remarkable performance playing grames (Silver et al., 2017).

Deep Reinforcement Learning (Deep RL) employs neural networks to manage environments with more complex state spaces, where traditional RL methods often fall short due to their limited ability to generalize from sparse data. In Deep RL, neural networks serve as the core architecture for both the policy mechanism and the value function approximator. These networks are particularly effective in complex environments as they excel at extracting and learning intricate patterns from high-dimensional data. In the realm of RL, no single algorithm universally fits all scenarios; each algorithm comes with its own set of trade-offs based on the specific characteristics of the environment (Sutton, 2024). Proximal Policy Optimization (PPO), however, has gained substantial popularity, especially in multi-agent settings. PPO enhances learning stability and performance by employing techniques such as gradient clipping and KL divergence (Schulman et al., 2017). Gradient clipping prevents the updates during training from being too large, which can disrupt the learning process; it effectively keeps the training steps small and manageable. KL divergence is a way to ensure that updates to the policy do not stray too far from what the agent has previously learned, promoting consistent and gradual improvement. Due to these features, PPO was the learning algorithm of choice in this paper's experiments.

Reproducibility remains a significant challenge in the field of reinforcement learning, particularly due to the variability in experimental setups and the stochastic nature of many RL algorithms (Henderson et al, 2018). The dependence on random conditions between simulations, algorithm hyperparameters, nondeterministic behavior, and even changes to software packages all open up the possibility of differing experimental outcomes. Updates to neural networks are accomplished with Stochastic Gradient Descent (SGD), which does not guarantee a unique outcome (Bottou et al., 2018). For most experiments in economics, results are only taken seriously once they have been reproduced many times over. The reproducibility portion of this paper is meant to authenticate the findings of Curry et al., and ensure their results were not by chance.

The traditional RBC model assumes that the representative consumer consumes according to a Constant Relative Risk Averse (CRRA) utility function

$$u(c) = \frac{c^{1-\eta}}{1-\eta}$$

where c represents consumption and η denotes the coefficient of relative risk aversion. From its Arrow-Pratt index we can understand that the degree of risk aversion present is dependent on the change in wealth of an individual. Although commonly used in the RBC model, it is not clear that it is always the best utility function for modeling the aggregate economy (Gali, 1994). Moreover, in an ACE model, as in Curry et al.'s, utilizing CRRA utility functions at a micro level may have consequences. After all, the use of this unchanging function may present criticism similar to that of formerly mentioned rule-based agents. Other utility functions are often used to model how people act when the outcomes of their actions are nondeterministic (Holt and Laury, 2002). Since the overall goal of agents is to mimic how humans may act and learn, it is worth considering such utility functions within an ACE model. For instance, consider the Constant Absolute Risk Aversion (CARA) utility function

$$u(c) = -e^{-\alpha c}$$

where c denotes consumption and α signifies the coefficient of absolute risk aversion. As evident from its Araprat-Index, this utility function keeps risk aversion constant with respect to changes in wealth. Additionally, as noted by Holt and Laury (2002), it is worth considering a hybrid of CRRA and CARA, which takes the following form

$$U(c) = \frac{1 - \exp(-\alpha c^{1-\eta})}{\alpha}$$

where c is the level of wealth or consumption, α is the parameter of absolute risk aversion, and η is the parameter of relative risk aversion. In the future, if a sophisticated ACE model were to be adopted into macroeconomic policy, it would have to consider such variation in consumption desire. Although using different utility functions within an objective can alter an agent's desire to consume, it shouldn't completely prevent the agent's learning ability. Complete failures to learn may point to more important underlying flaws in the reinforcement learning strategy. Such tests are conducted in this paper to gain a better understanding of the interactions in the environment.

The landscape of Rl with economics has seen a diverse array of applications, ranging from dynamic pricing in markets, as evidenced by Kutschinski et al.'s DMarks II platform (2003), to the complexities of fiscal policy with Chen et al.'s monetary model featuring household agents (2023). Additionally, addressing sequential social dilemmas, such as those studied by Leibo et al. with temporally-extended dilemmas (2017), illustrates the breadth of scenarios that RL approaches can capture. Given these sophisticated models, integrating RL into the RBC model represents a natural and promising progression to enhance our understanding of economic phenomena in multi-agent settings.

4. Experimental Design

The experiments in this paper are designed for 3 primary purposes. Firstly, to investigate if environmental observations align with what was originally seen by Curry et al. Secondly, to investigate whether or not agents fall into an approximate equilibrium. Lastly, to investigate if changing consumer utility functions has an unexpected effect on the integrity of outcomes in terms of both observed outcomes and/or the ability to find a collective equilibrium. The cumulative analysis of this paper's experiments will speak to the integrity of the multi-agent reinforcement learning representation of the RBC model.

In Analyzing micro-Founded general equilibrium models with many agents using deep reinforcement learning, a specific learning pattern is noted for the consumers, firms, and government. These are represented by their respective rewards over episodes (see Figure 16 in the appendix). The plotted rewards of each agent represent how well they optimize their respective objectives, as discussed below in experimental procedures. Ideally, agent rewards will plateau after many simulations, indicating that their valuations and policy mechanisms have converged. Failure to plateau generally indicates that their behavior is untrackable, which would make the interpretability of such a model unfruitful. Additionally, the output of an experiment will show the average prices, wages, and tax rates for performed simulations (see Figure 16 in the appendix). Likewise, if these metrics are unpredictably scattered, it could indicate another problem with the model's setup. The experiments used in this paper's reproducibility portion use the exact same input parameters as Curry et al. This includes using a CRRA function in the consumers' consumption decision problem. In the original experiments, 150000 training simulation episodes were conducted to investigate the previously mentioned metrics' convergence. In this paper's rendition, experiments were run for 200000 episodes to ensure the results in the original paper were not cut short of any unprecedented anomalies. The first part of the reproducibility challenge is satisfied if the convergence of important metrics is similar to those in the original experiments. They do not have to be identical, as certain parts of the experiment are nondeterministic (i.e., value updates with SGD). However, the general trajectory should be trackable and congruent with the original results.

The next part of the experiment, regarding the reproducibility challenge, analyzes whether the agents naturally fall into an ϵ-nash equilibrium (Curry et al, 2022). In other words, agents should only have a small incentive to deviate from their converged policies, represented by ϵ. As noted by Shoham et al (2007), finding an equilibrium in a multiagent setting is increasingly difficult in the number of present agents. Oftentimes, analytical solutions are impossible to calculate in multiagent sequential games. Recently, however, literature has started to recognize RL as a natural fit for such economic modeling, as it consists of rational agents maximizing their objectives in accordance with discounted future rewards (Haladane and Turrell, 2019). Empirical proof in gameplay has shown that RL agents can perform optimally in multi-agent sequential games (Vinyales et al, 2019). Such observations have fueled the likes of Curry et al. to consider extremely well-trained agents to be in an ϵ-nash equilibrium. It would be too careless to call this an exact equilibrium, given the nonconvexity of neural network loss landscape and certain non-deterministic aspects of the environment. Such features can render solutions non-unique or inaccurate. However, the paper does propose that the degree of rationality will be only offset by a theoretical constant: ϵ. The method for identifying equilibrium, known as best response training, is applied by selecting specific points from the list of training simulations. The process then unfolds in a series of steps for each agent: (1) All agent learning is frozen except for one; (2) The non-frozen agent's policy is then allowed to evolve through numerous training simulations; (3) Steps 1 and 2 are repeated until each agent type is covered. This iterative cycle tests whether agents feel the need to make policy changes for better reward streams. Further justification for this method is described by by Curry et al. (2022). Their original experiments found an approximate equilibrium; rerunning this experiment is necessary to validate such results.

Lastly, analysis regarding the convergence of metrics and the quality of ϵ-nash equilibrium is repeated using the formerly mentioned CARA and hybrid utility functions. This ablation study is used to understand how sensitive the learning method is to different underlying economic assumptions. Any non-interpretable anomalies may point toward how the experimental architecture could be changed to be more resilient.

5. Experimental Procedures

This section of the paper discusses the concept of rewards and walks through what happens across training simulations. To build up to this concept, a description of the environment and its variables is presented. An implemented learning curriculum is also described, which helps avoid trivially inaccurate equilibrium. The README files in the attached code repository provide a more technical description of how to run the code for the described experiments. Sections 5.1-4 are adapted from the original paper.

5.1 Agent Types

The model encompasses worker-consumers, price-setting firms, and a government that sets tax rates and redistributes income. Crucially, we do not assume market-clearing prices and wages at each timestep, allowing for potential over or under-demand of goods. A simulation in this economy consists of T=10 timesteps, and each timestep, t, represents a 3-month quarter in the economy. At each t, we simulate the following:

- Firms produce unique goods, each identified by an index $i \in I$, utilizing the labor of individuals who act both as worker and consumers;
- Consumers, indexed by $j \in J$, who work and consume goods;
- A singular government that sets tax rates on labor income and revenue from goods sales.

Each agent sequentially receives an observation $o_{i,t}$ of the world state s_t, takes an action $a_{i,t}$ derived from its policy π_i, and earns a reward $r_{i,t}$. The state of the environment, s, formally includes all states of the agents and a global state of the world. At each timestep t, all agents act simultaneously. However, certain actions impact the following timestep $t + 1$. For instance, the government sets tax rates to be applied at $t + 1$, visible to firms and consumer-workers at t, allowing them to adjust their policies accordingly. Likewise, at t, firms determine prices and wages effective at $t + 1$, influencing the global state in the subsequent timestep. This arrangement resembles but does not fully replicate, a Stackelberg leader-follower dynamic, where the government (leader) acts first, followed by firms and consumers (followers), as described by von Stackelberg et al. (2010). Typically, this sequence grants followers a strategic benefit by providing them with additional information to guide their decision-making.

5.2 Consumer-Workers

At each timestep t, individual j works $l_{j,t}$ hours and consumes $c_{i,j,t}$ units of good i, choosing to work for a specific firm i at each timestep. While consumer-workers attempt to consume $\hat{c}_{i,j,t}$, each good, priced at p_i by the firms, is subject to the government-set income tax rate. Consumers cannot exceed their budget; if attempted consumption costs surpass their budget, consumption is scaled so that $\sum_i p_i \hat{c}_{t,i,j} = B_j$. Actual consumption is contingent on available goods inventory $y_{i,t}$, with total demand for good i being $\hat{c}_{i,t} = \sum_j \hat{c}_{i,j,t}$. If supply is insufficient, goods are rationed proportionally:

$$c_{i,j,t} = \min\left(1, \frac{y_{i,t}}{\hat{c}_{i,t}}\right) \hat{c}_{i,j,t}.$$

Working and consuming alter a consumer's budget $B_{j,t}$. A consumer earns labor income $z_{j,t} = \sum_i l_{i,j,t} w_{i,t}$, with each firm paying a wage $w_{i,t}$. Consumption costs are $\sum_i p_{i,t} \cdot c_{i,j,t}$, and with tax rate τ_t, income tax paid is $\tau_t \cdot z_{j,t}$. Total tax revenue R_t, including taxes from firms, is redistributed evenly among workers. Hence, the budget update is:

$$B_{t+1,j} = B_{j,t} + (1 - \tau_t) z_{j,t} + \frac{R_t}{|J|} - \sum_i p_{i,t} \cdot c_{i,j,t}.$$

Each consumer aims to maximize utility, defined as:

$$\max \pi_j \mathbb{E}_{c,l \sim \pi_j} \left[\sum_t \gamma_t^c \sum_i u(c_{i,j,t}, l_{i,j,t}, \theta_j) \right],$$

where the entire utility function $u(c, l, \theta)$ is given by:

$$u(c, l, \theta) = \frac{(c+1)^{1-\eta} - 1}{1 - \eta} - \frac{\theta}{2} l^2,$$

with γ_c as the consumer's discount factor. The left term is how the consumer objective integrates isoelastic utility (CRRA). The η value is set to 0.1 across all CRRA experiments in this paper, as it was used in the original paper; with this small positive value, marginal utility declines with more consumption, but not too dramatically. An additional -1 is added at the end of the numerator of this for computational convenience. It ensures that the term does not become 0 when consumption is 0; however, it does not change the nature of CRRA.

The right term of the function represents the disutility of work, with coefficient θ_j varying among workers according to a distribution of very small positive decimals. These values are annealed over successive time steps. The functional form of the disutility of labor term remains constant even in the later experiments when the right term is altered. In the later experiment, which implements CARA for the utility of consumption, the entire utility function is defined as:

$$u(c, l, \theta) = -e^{-\alpha c} - \frac{\theta^2}{2} l^2$$

and the α value is set to 0.5. As used by Blavatskyy (2022), this alpha value represents a reasonable level of risk aversion, independent of the level of wealth, and also exhibits diminishing marginal utility. A typically high value was chosen to apply more pressure to the model for sensitivity analysis. It is worth noting that such experiments will have negative utility due to the functional form. However, this does not translate to a failure in the experiment; all that matters is that the transitive ordering of the utility remains reliable. In the experiment with the hybrid between CARA and CRRA utility consumption functions, the entire consumer utility function takes the form:

$$u(c, l, \theta) = \frac{1 - \exp(-\alpha c^{1-\eta})}{\alpha} - \frac{\theta^2}{2} l^2$$

In the code, certain adjustments are made to prevent underflow; however, these changes don't undermine the integrity of the function. The parameters are set to 0.1 and 0.5 for η and α, respectively, to gain exposure to mixed extremes between the 2 formerly mentioned experiment configurations.

5.3 Firms

At every timestep t, a firm engages in several activities: it receives labor from workers, produces and sells goods, and may decide to invest in its capital. Each firm sets a price $p_{t+1,i}$ and wage $w_{t+1,i}$ for the next timestep $t + 1$. Capital investment by a firm increases its capital stock according to the formula $k_{t+1,i} = k_{t,i} + \Delta k_{t,i}$.

Firms use their available capital $k_{t,i}$ and the total labor $L_{t,i}$ to produce goods. The output of goods $Y_{t,i}$ is determined by a production function

$$Y_{t,i} = A_i k_{t,i}^{1-\alpha} L_{t,i}^{\alpha}$$

, where α (ranging from 0 to 1) represents the relative importance of capital versus labor in the production process. Note that this is a different α than what was used for implementing CARA.

Consumers purchase $C_{t,i}$ units of each good, affecting the firm's inventory, which updates as $y_{t+1,i} = y_{t,i} + Y_{t,i} - C_{t,i}$. Inventories remain positive, indicating that only produced goods are available for consumption. Firms earn profits or incur losses through their operations, calculated by

$$P_{t,i} = p_{t,i} C_{t,i} - w_{t,i} L_{t,i} - \Delta k_{t,i}$$

. In the latter equation, w's represent the wages paid, which are set as discussed in the Consumer-Workers section. L represents labor hours of workers. The firm's budget for the next period is then updated to

$$B_{t+1,i} = B_{t,i} + (1 - \sigma_t) P_{t,i}$$

, where σ_t is the corporate tax rate paid to the government. While firms can incur debt temporarily, they are required to maintain a non-negative budget at the end of each episode, adhering to the no-Ponzi condition. This stipulation promotes sustainable investment strategies aimed at enhancing future economic growth. Each firm aims to maximize its profit, optimizing its operational decisions based on economic parameters and regulations. This goal is mathematically represented by the optimization problem:

$$\max_{\pi_i} \mathbb{E}_{p,w,\Delta k \sim \pi_i} \left[\sum_t \gamma_t^f P_{t,i}(p_{j,t}, w_{j,t}, \Delta k_{j,t}) \right],$$

where γ_t^f is the firm's discount factor, influencing the valuation of future profits relative to current profits.

5.4 Government

The government, also referred to as the social planner and indexed by p, is responsible for setting corporate and income tax rates. It collects total tax revenue $R_t = \sigma_t \sum_j z_{j,t} + \tau_t \sum_i P_{i,t}$, where σ_t and τ_t are the corporate and income tax rates, respectively. This collected revenue is uniformly redistributed among the consumer-workers, keeping the government's budget balanced at zero.

To optimize social welfare, the government formulates its policy π_p as follows:

$$\max_{\pi_p} \mathbb{E}_{\tau,\sigma\sim\pi_p}\left[\sum_t \gamma_t^p \operatorname{swf}(\tau_t, \sigma_t, s_t)\right],$$

where $\operatorname{swf}(s_t)$ represents the social welfare at timestep t, and γ_p is the discount factor applied by the government. For these experiments, social welfare is defined as total consumer utility after redistribution. This definition can drastically change outcomes, but for the sake of reproducibility, it is kept as is.

5.5 Rewards

As previously shown, each agent chooses which action to take in accordance with their optimized objective over the future expected stream of timesteps. Once an action is chosen that maximizes such an objective, this action is used to calculate the the reward for time t. So simply put, at time t the reward that consumer j receives is their chosen consumption and labor plugged into

$$R_{j,t} = u(c_{j,t}, l_{j,t}) = U(c) - \frac{\theta^2}{2} l^2$$

where, U(c) can take the form of CRRA, CARA, or a hybrid depending on the experiment. By the same token firm i gets a reward at time t that can be calculated with their units of each good they sold C, the price of each good they sold p, their change in capital stock

Δk, the wage paid to each worker w, and the hours dedicated to labor for each worker L. All such variables are then plugged into the equation below.

$$R_{i,t} = P_{i,t}(p, k, w, L,) = p_{t,i}C_{t,i} - w_{t,i}L_{t,i} - \Delta k_{t,i}$$

Similarly, reward for the government at time t is simply the summed-up consumer utility noticed at said time period. In summary, the rewards at time t are what was actually noticed in terms of consumer utility, firm profit, and social welfare. The rewards are then scaled by 5 for consumers, 30000 for firms, and 1000 for governments to emphasize the vast differences between entities in the real world.

5.6 Running

As in the original paper, these experiments were run with 100 consumers, 10 firms, and 1 government. In both papers, the episodes were each 10 time steps (quarters) long. In the original paper, the training period lasted 150000 episodes; the experiments in this paper were 200000 episodes long to also ensure convergence. This paper repeats the 200000 episode training trials twice to ensure reproducible results. The original paper reproduced trials 3 times over (but had much more funding to do so). In these trials, every 2000 episodes, information is saved regarding model parameters and metrics describing the environment. After training experiments are run with CARA, CRRA (as in the original paper), and the hybrid function, the post hoc best response training is conducted as described in the Experimental Design section. In this best response training, each agent was allowed to train their policies for 25000 episodes, as all other agents had their policy training frozen. This process saves any rewards gained from using updated and old policies throughout the 25000 simulations. Best response training experiments were conducted at episode 100000 in the middle of the trial and at episode 200000 at the end of the trial. Testing the before and after provides insight into how the approximated equilibrium changes. More specifics about the input parameters to the model, such as initial capital distributions, firm endowments, prices and much more can be found in the appendix (Figures 1, 2, and 3). All of the experiments in this paper set the economy open, which conditionally allows firms to sell goods to imaginary agents outside of the economy. A further explanation of this open market concept is presented in the appendix (Figure 4).

5.7 Exploration and Training Curriculum

To make sure agents explore the state space adequately and become aware of the possibility of rewards from differing actions, a degree of entropy is introduced to agent policies. This value is annealed over time to eventual negligence; however, in the nascent stage of learning, it does play a role in increasing the number of possible states from which an agent may acquire knowledge. Such practice is typical in RL (Sutton 2024). Additionally, a training curriculum is employed within the experiments to make sure agents can learn past their original behaviors. In this curriculum, consumers start training their policies right away. Thereafter, at episode 30000, the firms start training theirs. Finally, in episode 60000, the government starts training its policy. This curriculum is akin to the fact that people in real life enter the job market with at least some prior knowledge about the tradeoff between work and leisure. If the experiments were to throw consumers into the economy before learning about how the environment works in a more stationary manner, the trials would fail to produce interpretable results. Without nascent exploration, the agents wouldn't be sufficiently aware of how their actions may affect the future. Such learning curriculums are also common in RL (Bengio et al., 2009). The image below illustrates the progression of the curriculum.

As mentioned earlier, the neural networks behind agent policy and value functions are updated throughout the PPO algorithm. A more in-depth analysis of this process is shown in the appendix (Figures, 6, 7, and 8).

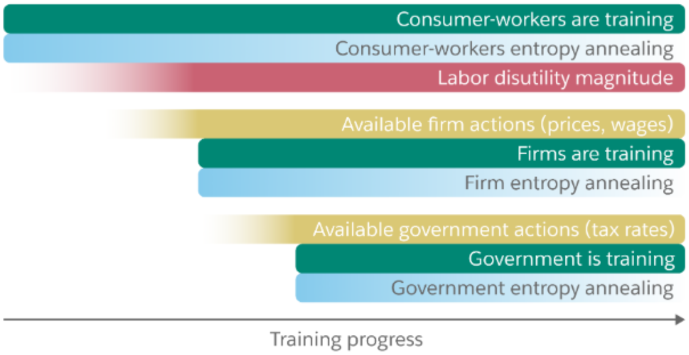

Consumer-workers are training
Consumer-workers entropy annealing
Labor disutility magnitude
Available firm actions (prices, wages)
Firms are training
Firm entropy annealing
Available government actions (tax rates)
Government is training
Government entropy annealing
Training progress

Figure 1: Figure 5: Training Curriculum

6. Results

The following subsections present the results of the experiments. The first subsection (CRRA Results) focuses on the same experiments conducted in Curry et al., with isoelastic utility for consumption. The next subsections present the findings of the ablation studies in which CARA and hybrid consumption utility functions were used. All experiments consider the progression of environment observations and the status of equilibrium over the course of training simulations. Trials for a given subsection resulted in similar convergence, so each subsection displays a representative trial for its respective experiment.

6.1 CRRA Results

The following graphs show the progression of rewards for consumers and firms throughout a single 200000 episode learning trial. Rewards are averaged over their respective agent types, and the 25th and 75th percentile range is shown.

As we can see, the learning process for firms and consumers in the shown trial is very similar to the runs presented in the original paper (Figure 16 in the appendix). For consumers, there are dips at the beginning, especially as firms and the government just begin their training at episode 30000 and 60000, respectively. This is predictable because, at such points, the firms and consumer policies are somewhat random, implying that wages and prices might be set too high and a balance of social welfare isn't recognized. Eventually, the

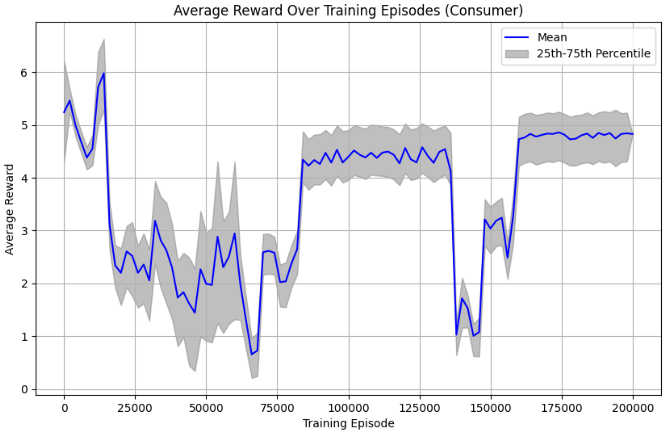


Figure 9: Consumer Rewards with CRRA

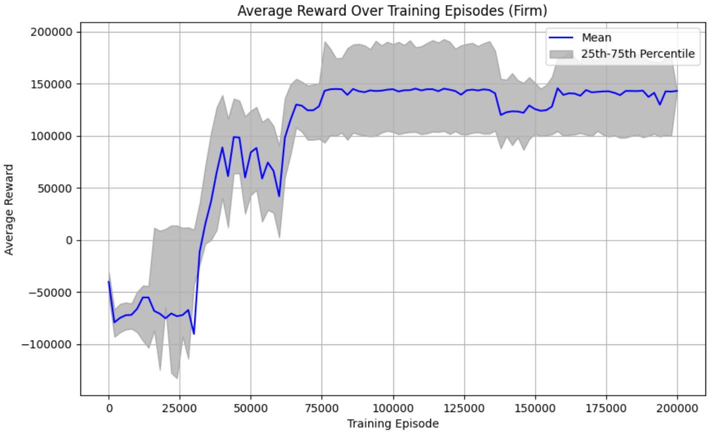


Figure 10: Firm Rewards with CRRA

trajectory becomes uphill. In both papers, a somewhat concerning dip happened around 125000-150000 episodes in. In the original paper, this coincides with the end of the trial. However, in this paper's trial, because it was 200000 episodes long, a rebound is visible back to the plateaued point. Additionally, firm rewards plateau and the variance of rewards for individual agents is relatively controlled. It is reasonable to conclude these agents were able to refine their original policies.

The graphs below show the progression of prices and wages in the economy, throughout the training trial.

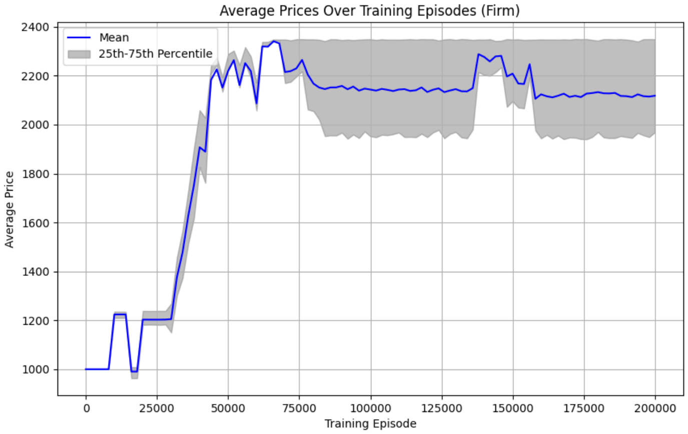


Figure 11: Prices with CRRA

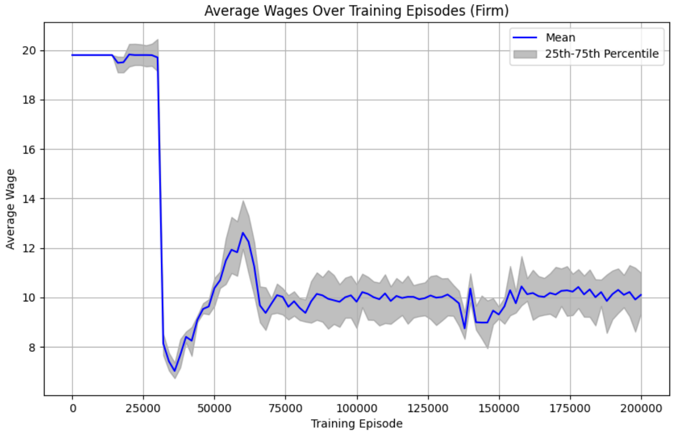


Figure 12: Wages with CRRA

Again, these results are very similar to those from the original paper (Figure 16 in the appendix). A positive correlation between firm rewards and their ability to increase prices can be noted. Just after the decrease in wages as a result of firms starting to train their policy, wages eventually start to rise again. This does not hinder their reward progression, as more consumers are willing to work with the increase in wages, leading to more output from the firms.

The following graphs show the progression of government rewards (total consumer consumption after distribution). As well, they show the progression of corporate and income tax rates as percentages. In this economy there is only one government, so the averages are analogous to the actual rates.

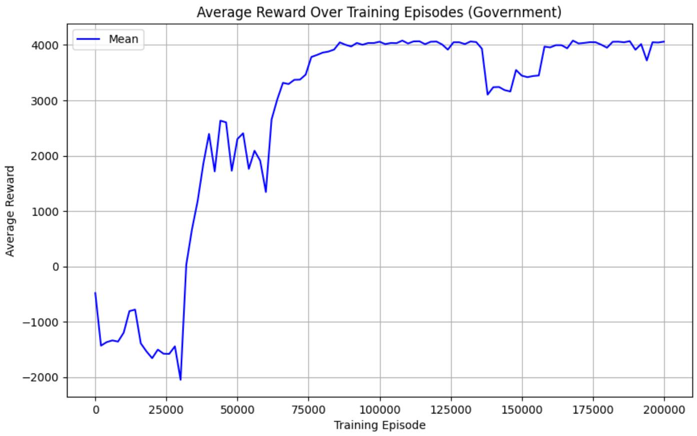


Figure 13: Social Welfare with CRRA

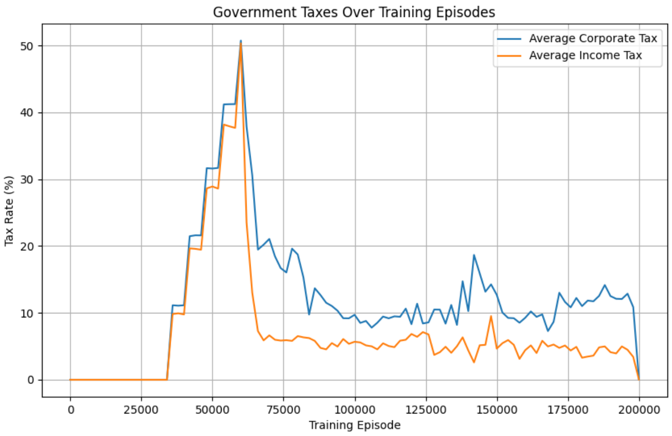


Figure 14: Taxes with CRRA

These graphs were not produced in the original paper; however, they add to the analysis of the economy and provide a base comparison for future experiments. One important observation is that the government is not incentivized to over-tax in order to get closer to its goal of maximizing consumption after distribution. In fact, directly after the government starts training its policy, it drastically reduces both the corporate and income tax rates. This points to their inability to scale the redistribution of wealth. There are, of course, some taxes still, as it can manage to redistribute smaller quantities. Overall, training for the government is also steady.

The following graph displays the status of equilibrium. The bars associated with the key word ”middle” are results from the best response training at episode 100000. The bars associated with the key word ”end” are results from the best response training at episode 200000. Blue lines represent the mean rewards achieved from the simulated 25000 best response episodes with the current policy, freezing all other agent training. The purple bars show the mean rewards achieved from the simulated 25000 best response episodes with a trainable policy, freezing all other agent training. Mean rewards for firms and the government are measured in units of 10^4 and 10^3, respectively.

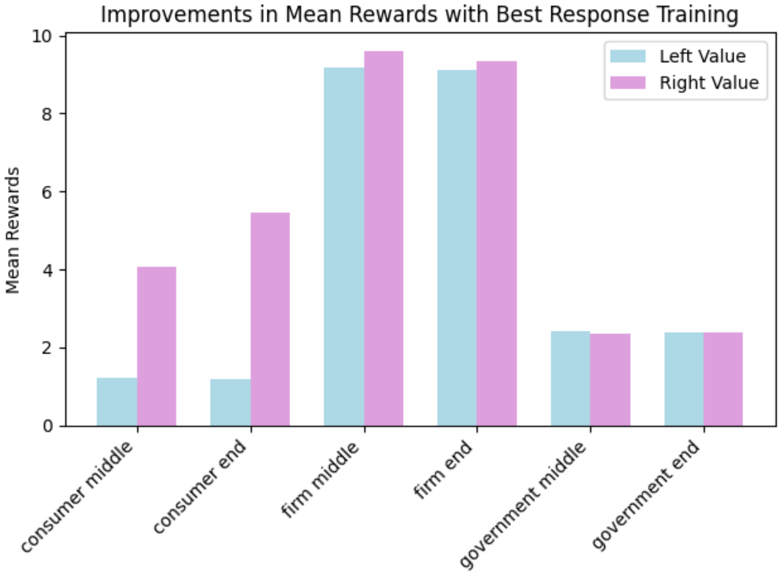


Figure 15: Equilibrium Analysis with CRRA

Firms and the government appear to have very little motivation to deviate from their policies; the small incentive diminishes even more by the end of the training. This is similar to the original paper’s findings, as shown in Figure 17 in the appendix. However, the consumer’s story seems to be a bit different. The original findings show that at episode 100000, consumers are not in equilibrium; however, at episode 150000, they are. On the other hand, in this trial, they are not at an equilibrium at any point. This can be a byproduct of allowing the training period to last for 50000 more episodes. In fact, as noted in Figure 9, the policy indeed changed within these extra episodes. This could indicate that the consumers need more time to learn their optimal policies before we can conclude they are in equilibrium. Future experiments with such models may even consider doubling the training period for analysis.

Overall, The behavior of agents in this model can be interpretable. A large portion of this behavior is reproducible with respect to the original paper; however, consumer policies have not truly converged under such experiments. The government and firms in this model do find their approximate equilibrium, so further changes to the model's experiments should focus on the consumer-workers.

6.2 CARA

The following graphs show the progression of rewards for consumers and firms throughout a single 200000 episode learning trial. Rewards are averaged over their respective agent types, and the 25th and 75th percentile range is shown.

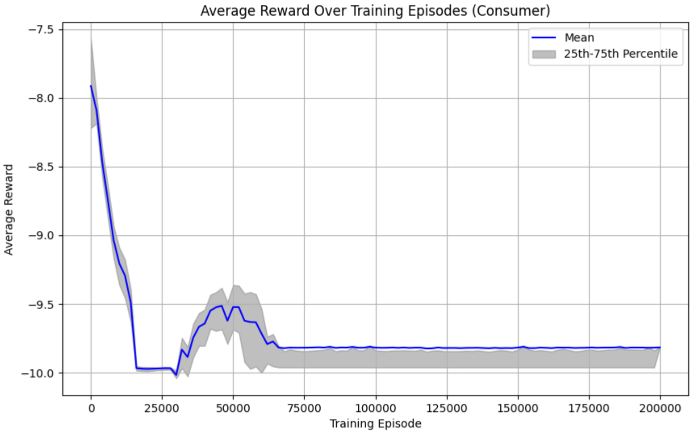


Figure 18: Consumer Rewards with CARA

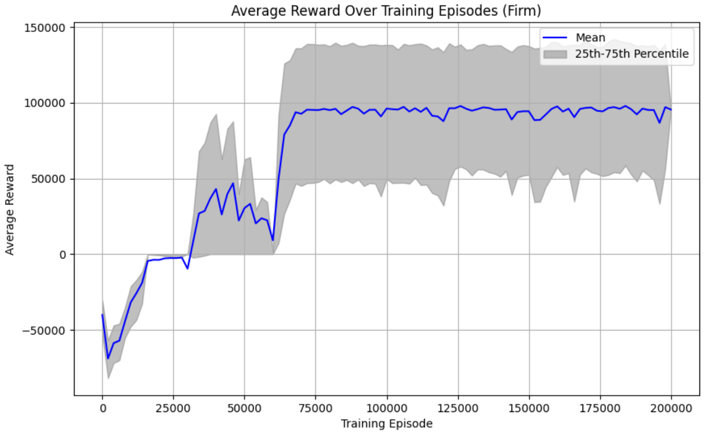


Figure 19: Firm Rewards with CARA

The convergence of mean rewards for consumers and firms is much more pronounced in this experiment with the CARA utility function. This is likely due to the fact that with CARA, risk aversion is not dependent on wealth as it is in CRRA. Consequently, the utility received from consumption is more rigid between different episodes in the simulation. Very soon after the government starts training, a steady state for consumer rewards is found.

As this, in turn, affects the number of labor hours, we can see firms also follow suit with a solidified employment pool.

The graphs below show the progression of prices and wages in the economy throughout the training trial.

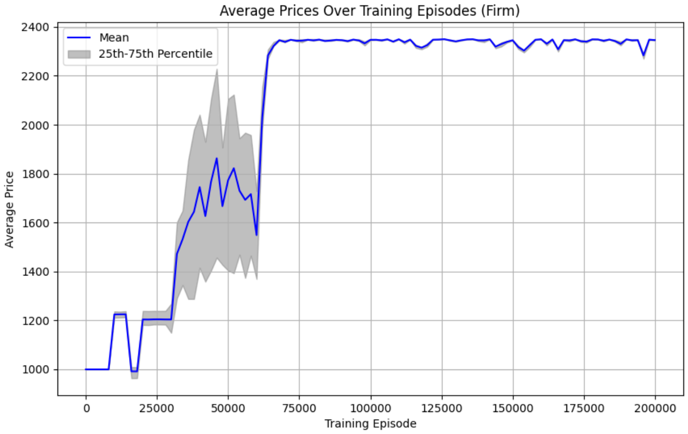


Figure 20: Prices with CARA

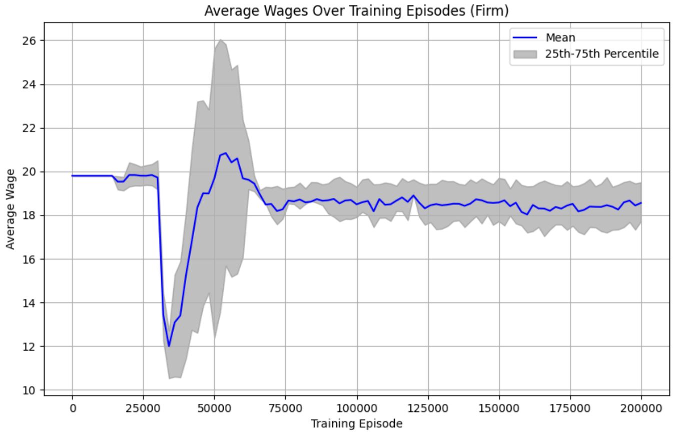


Figure 21: Wages with CARA

The convergence of average wages and prices also follows what would be expected from employing a CARA function for consumer utility. One interesting observation is that average wages with CARA are significantly higher than they are with CRRA. In economics literature, it has been noted that with greater degrees of risk aversion, people generally require more compensation than the expected value over uncertainty would predict. Likewise, in this experiment, workers collectively demand higher wages to further ensure they can purchase goods, under the uncertainty that such goods may change in price. Furthermore, with agents acting as such, the prices of goods converge faster as well.

The following graphs show government rewards and taxes as in the CRRA experiment. These graphs, using CARA, are fairly similar in trajectory to those produced with CRRA. Corporate tax still exceeds income tax, and the two graphs converge slightly better. As in the previous section, the bar graph represents the best response training results for an equilibrium analysis.

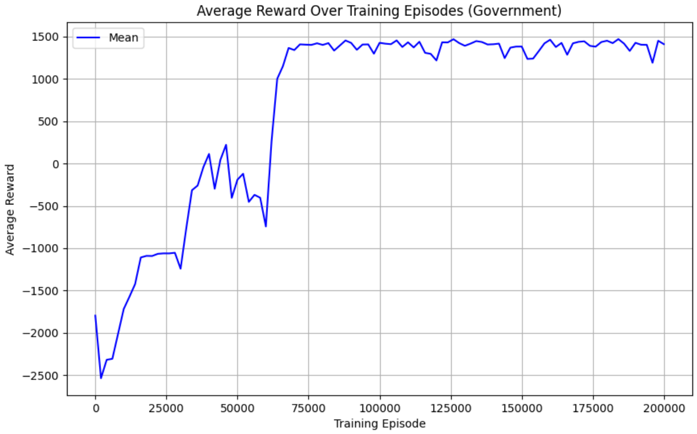


Figure 22: Social Welfare with CARA

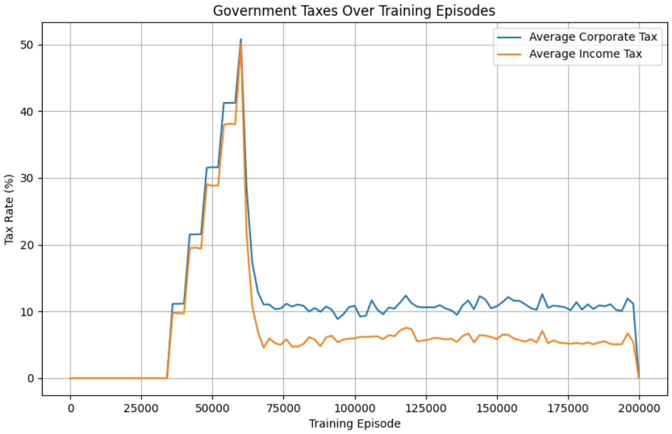


Figure 23: Taxes with CARA

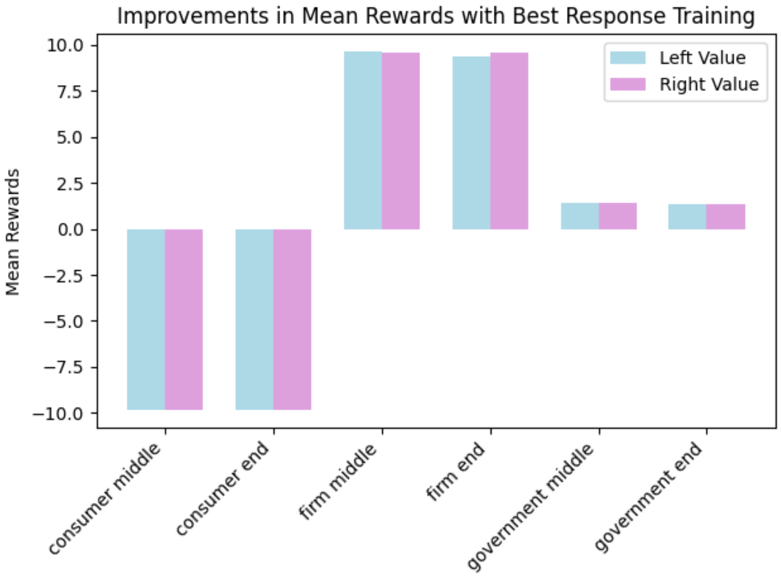


Figure 24: Equilibrium Analysis with CARA

Amazingly, At both the middle and the end, all agents had almost no reason to deviate from their learned policy. These results follow the more rigid convergence that was seen across previous figures in the section.

Equipping the consumer workers in the economy with CARA utility functions leads to much more trackable results than what was shown in the CRRA section. All agents seemed to be at an approximate equilibrium, and this state was found much quicker. However, in terms of mapping this model to the real world, a CARA utility function may be too rigid of an assumption to make regarding a population. Further testing with other utility functions and different attached parameters may be necessary in the future.

6.3 HYBRID

The following graphs show the progression of rewards for consumers and firms throughout a single 200000 episode learning trial, with hybrid utility functions. Rewards are averaged over their respective agent types, and the 25th and 75th percentile range is shown.

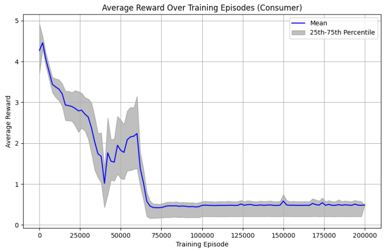


Figure 25: Consumer Rewards

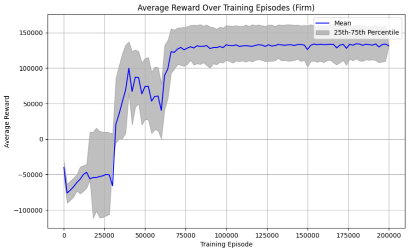


Figure 26: Firm Rewards

The graphs above show a stronger learning convergence than basic CRRA; this is predictable as the use of a hybrid function takes after many aspects of CARA.

The following graphs show the prices and wages observed in the experiment with a hybrid utility function.

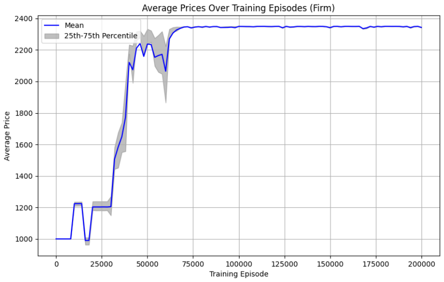


Figure 27: Prices with HYBRID

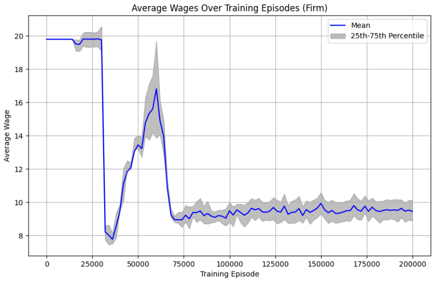


Figure 28: Wages with HYBRID

Interestingly, despite the previously noted strong convergence, the results of this hybrid experiment produce wages and prices much more alike to those seen in the original experiment, which used CRRA. This could indicate that the hybrid function is more optimal for training the RL agents. On one hand, the weight from CARA helps with convergence, and on the other hand, the weight from CRRA allows the environment observations to develop as such.

The following graphs show government rewards and taxes as in previous sections.

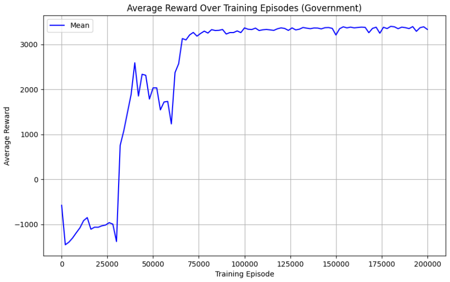


Figure 29: Social Welfare with HYBRID

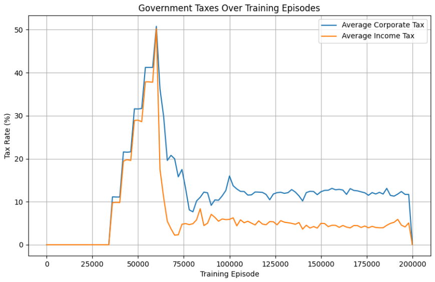


Figure 30: Taxes with HYBRID

As with the graphs previously displayed, the results are close to what was shown in the original paper using CRRA.

As in the previous section, this bar graph represents the best response training results for an equilibrium analysis. Much like CARA, these results show agents to all be at an approximate equilibrium, with little to no incentive to change their policies.

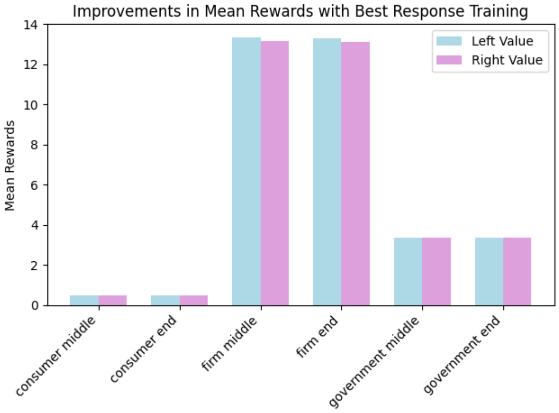


Figure 13: Figure 31: Equilibrium Analysis with HYBRID

7. Conclusion

This paper probed the findings of Curry et al., which originally used a CRRA utility function as the backbone for consumer rewards throughout an RL learning curriculum. In conclusion, the original results from Analyzing micro-founded general equilibrium Models with many agents using deep reinforcement Learning are reproducible for the most part. However, upon allowing training sessions to carry out for more than the original 150000 episodes, the behavior of consumer-workers fell out of the proposed ϵ-nash equilibrium. This raises questions regarding how experiments should be conducted in the future. In Deep RL, given the use of neural networks, it is possible for policies to jump around every so often. For this reason, increasing the learning period may be necessary to get reliable results. Aside from consumers, firms and the government seemed to find their optimal strategies.

Additionally, this paper investigated the effects of changing the consumption rewards to conform to CARA and hybrid utility functions. The results of both seemed to do a better job than CRRA in finding a proposed ϵ-nash equilibrium for all agents. The results found came much quicker following more abrupt convergence. However, CARA has been adopted less in macroeconomic models due to its more rigid assumption that the degree of wealth doesn't impact risk aversion. Luckily, however, the hybrid utility function that inherits qualities from both CARA and CRRA, was able to produce similar environmental results to the original paper's experiments, all while converging elegantly. The results were quite interpretable for all 3 types of experiments conducted. The changes in agent behavior were in line with how economics traditionally expects humans to behave under such reward functions; this could point to a promising future for RL in ACE.

Although quite interesting, the model investigated in this paper is far from perfect. Rather, it is a pioneering initiative as to how multiagent models may look with the incorporation of reinforcement learning. The contributions of this paper are broadly meant to help understand what happens upon changing economic assumptions; and, how doing so can impact the rationality of model microfoundations. With access to more computing power, further experiments can test different utility function parameters or alter the simulation to allow different consumers to have differing utility functions within the same experiment. Additionally, tests with other environment initialization may help prove such models more reproducible.

8. Bibliography

A. G. Haldane and A. E. Turrell, "Drawing on Different Disciplines: Macroeconomic Agent-Based Models," *Journal of Evolutionary Economics*, vol. 29, no. 1, pp. 39–66, March 1, 2019. `https://doi.org/10.1007/s00191-018-0557-5`.

B. Heer and A. Maußner, *Dynamic General Equilibrium Modeling*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2009. `https://doi.org/10.1007/978-3-540-85685-6`.

P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep Reinforcement Learning That Matters," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, April 29, 2018. `https://doi.org/10.1609/aaai.v32i1.11694`.

C. A. Holt and S. K. Laury, "Risk Aversion and Incentive Effects," *The American Economic Review*, vol. 92, no. 5, pp. 1644–1655, 2002.

A. P. Kirman, "Whom or What Does the Representative Individual Represent?," *Journal of Economic Perspectives*, vol. 6, no. 2, pp. 117–136, June 1992. `https://doi.org/10.1257/jep.6.2.117`.

E. Kutschinski, T. Uthmann, and D. Polani, "Learning Competitive Pricing Strategies by Multi-Agent Reinforcement Learning," *Journal of Economic Dynamics and Control*, vol. 27, pp. 2207–2218, September 1, 2003. `https://doi.org/10.1016/S0165-1889(02)00122-7`.

J. Z. Leibo, V. Zambaldi, M. Lanctot, J. Marecki, and T. Graepel, "Multi-Agent Reinforcement Learning in Sequential Social Dilemmas," *arXiv.org*, February 10, 2017. Available online: `https://arxiv.org/abs/1702.03037v1`.

J. Lintner, "The Valuation of Risk Assets and the Selection of Risky Investments in Stock Portfolios and Capital Budgets," *The Review of Economics and Statistics*, vol. 47, no. 1, pp. 13–37, 1965. `https://doi.org/10.2307/1924119`.

R. E. Lucas, "Econometric Policy Evaluation: A Critique," *Carnegie-Rochester Conference Series on Public Policy*, vol. 1, pp. 19–46, January 1, 1976. `https://doi.org/10.1016/S0167-2231(76)80003-6`.

“Optimization Methods for Large-Scale Machine Learning,” *SIAM Review*, Accessed April 30, 2024. https://epubs.siam.org/doi/10.1137/16M1080173.

J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal Policy Optimization Algorithms,” *arXiv.org*, July 20, 2017. Available online: https://arxiv.org/abs/1707.06347v2.

A. Shaikh, *Capitalism: Competition, Conflict, Crises*, Oxford University Press, 2016. https://doi.org/10.1093/acprof:oso/9780199390632.001.0001.

Y. Shoham, R. Powers, and T. Grenager, “If Multi-Agent Learning Is the Answer, What Is the Question?,” *Artificial Intelligence*, vol. 171, no. 7, pp. 365–377, 2007. Available online: https://www.google.com/search?q=Shoham%2C+Y.%2C+Powers%2C+R.%2C+%26+Grenager%2C+T.+(2007).+%22If+multi-agent+learning+is+the+answer%2C+what+is+the+question%3F%22+Artificial+Intelligence%2C+171(7)%2C+365-377.

D. Silver, J. Schrittwieser, K. Simonyan, I. Antonoglou, A. Huang, A. Guez, T. Hubert, et al., “Mastering the Game of Go without Human Knowledge,” *Nature*, vol. 550, pp. 354–359, October 1, 2017. https://doi.org/10.1038/nature24270.

H. von Stackelberg, *Market Structure and Equilibrium*, Springer Science & Business Media, 2010.

“Sutton & Barto Book: Reinforcement Learning: An Introduction,” Accessed March 11, 2024. http://incompleteideas.net/book/first/the-book.html.

L. Tesfatsion, “Agent-Based Computational Economics: Growing Economies from the Bottom Up,” *Artificial Life*, vol. 8, no. 1, pp. 55–82, 2002. https://doi.org/10.1162/106454602753694765.

C. R. Tilbury, “Reinforcement Learning for Economic Policy: A New Frontier?,” *arXiv*, February 23, 2023. Available online: http://arxiv.org/abs/2206.08781.

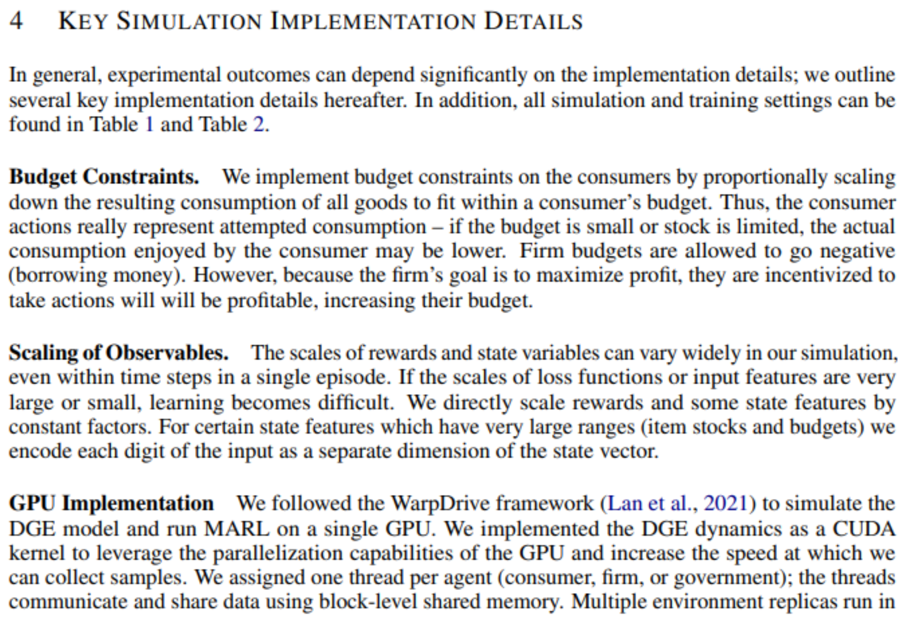
O. Vinyals, I. Babuschkin, W. M. Czarnecki, M. Mathieu, A. Dudzik, J. Chung, D. H. Choi, et al., “Grandmaster Level in StarCraft II Using Multi-Agent Reinforcement Learning,” *Nature*, vol. 575, no. 7782, pp. 350–354, November 2019. https://doi.org/10.1038/s41586-019-1724-z.

S. Zheng, A. Trott, S. Srinivasa, N. Naik, M. Gruesbeck, D. C. Parkes, and R. Socher, "The AI Economist: Improving Equality and Productivity with AI-Driven Tax Policies," *arXiv*, April 28, 2020. Available online: https://doi.org/10.48550/arXiv.2004.13332.

S. Zheng, A. Trott, S. Srinivasa, D. C. Parkes, and R. Socher, "The AI Economist: Optimal Economic Policy Design via Two-Level Deep Reinforcement Learning," *arXiv*, August 5, 2021. Available online: https://doi.org/10.48550/arXiv.2108.02755.

S. Zheng, Y. Yue, and J. Hobbs, "Generating Long-Term Trajectories Using Deep Hierarchical Networks," in *Advances in Neural Information Processing Systems*, vol. 29, Curran Associates, Inc., 2016. Available online: https://proceedings.neurips.cc/paper/2016/hash/fe8c15fed5f808006ce95eddb7366e35-Abstract.html.

9. Appendix



4 Key Simulation Implementation Details

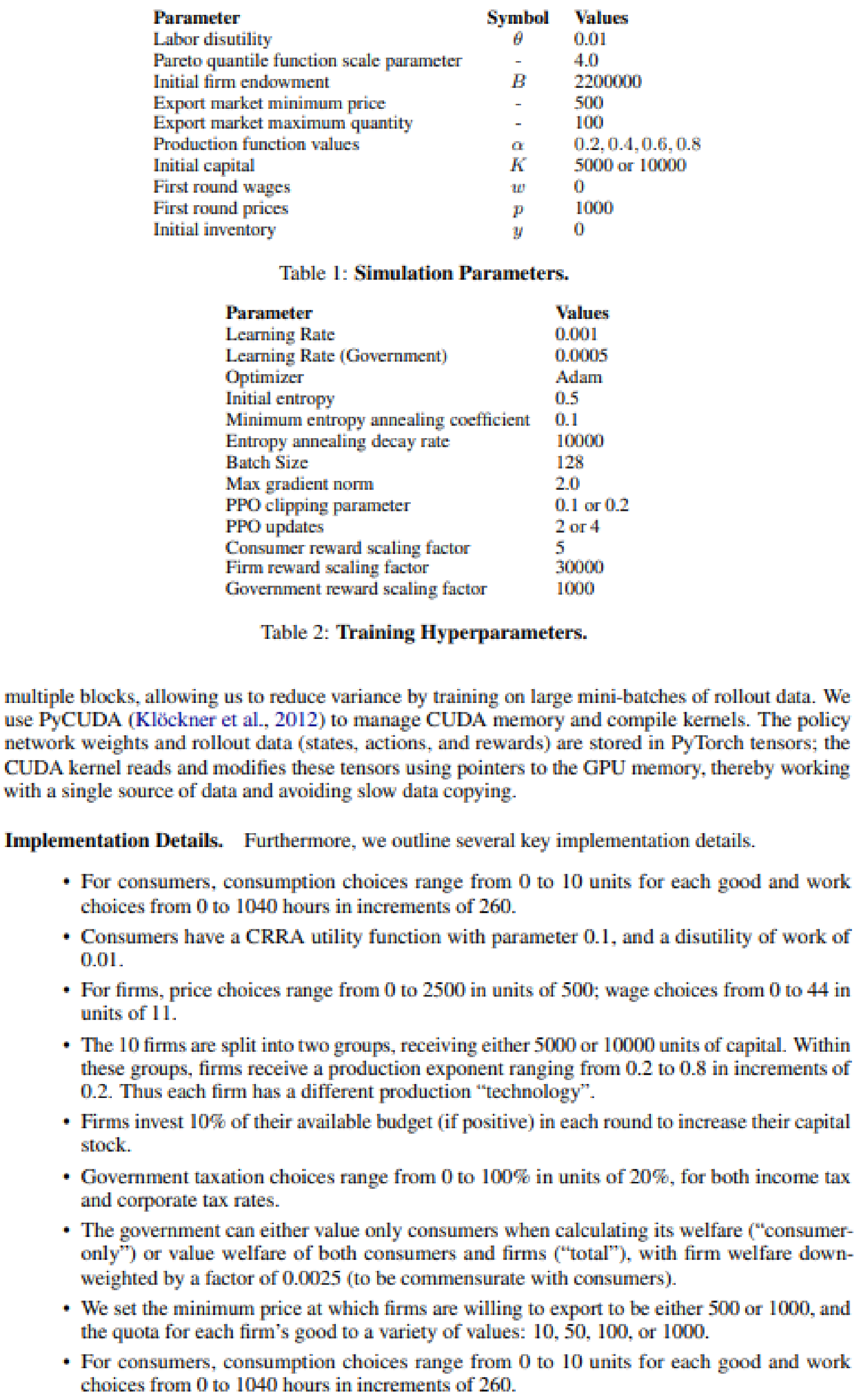
In general, experimental outcomes can depend significantly on the implementation details; we outline several key implementation details hereafter. In addition, all simulation and training settings can be found in Table 1 and Table 2.

Budget Constraints. We implement budget constraints on the consumers by proportionally scaling down the resulting consumption of all goods to fit within a consumer's budget. Thus, the consumer actions really represent attempted consumption – if the budget is small or stock is limited, the actual consumption enjoyed by the consumer may be lower. Firm budgets are allowed to go negative (borrowing money). However, because the firm's goal is to maximize profit, they are incentivized to take actions will will be profitable, increasing their budget.

Scaling of Observables. The scales of rewards and state variables can vary widely in our simulation, even within time steps in a single episode. If the scales of loss functions or input features are very large or small, learning becomes difficult. We directly scale rewards and some state features by constant factors. For certain state features which have very large ranges (item stocks and budgets) we encode each digit of the input as a separate dimension of the state vector.

GPU Implementation We followed the WarpDrive framework (Lan et al., 2021) to simulate the DGE model and run MARL on a single GPU. We implemented the DGE dynamics as a CUDA kernel to leverage the parallelization capabilities of the GPU and increase the speed at which we can collect samples. We assigned one thread per agent (consumer, firm, or government); the threads communicate and share data using block-level shared memory. Multiple environment replicas run in

Figure 1: Key Simulation Implementations and Details



Parameter	Symbol	Values
Labor disutility	θ	0.01
Pareto quantile function scale parameter	-	4.0
Initial firm endowment	B	2200000
Export market minimum price	-	500
Export market maximum quantity	-	100
Production function values	α	$0.2, 0.4, 0.6, 0.8$
Initial capital	K	5000 or 10000
First round wages	w	0
First round prices	p	1000
Initial inventory	y	0

Table 1: **Simulation Parameters.**

Parameter	Values
Learning Rate	0.001
Learning Rate (Government)	0.0005
Optimizer	Adam
Initial entropy	0.5
Minimum entropy annealing coefficient	0.1
Entropy annealing decay rate	10000
Batch Size	128
Max gradient norm	2.0
PPO clipping parameter	0.1 or 0.2
PPO updates	2 or 4
Consumer reward scaling factor	5
Firm reward scaling factor	30000
Government reward scaling factor	1000

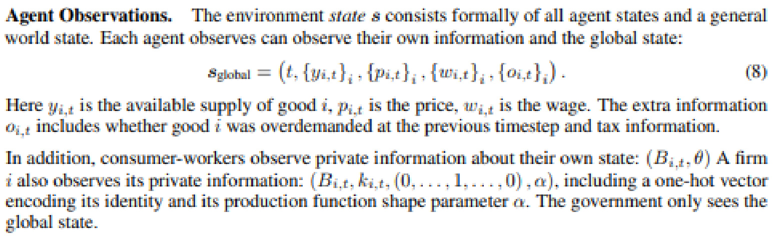
Table 2: **Training Hyperparameters.**

multiple blocks, allowing us to reduce variance by training on large mini-batches of rollout data. We use PyCUDA (Klöckner et al., 2012) to manage CUDA memory and compile kernels. The policy network weights and rollout data (states, actions, and rewards) are stored in PyTorch tensors; the CUDA kernel reads and modifies these tensors using pointers to the GPU memory, thereby working with a single source of data and avoiding slow data copying.

Implementation Details. Furthermore, we outline several key implementation details.

- For consumers, consumption choices range from 0 to 10 units for each good and work choices from 0 to 1040 hours in increments of 260.
- Consumers have a CRRA utility function with parameter 0.1, and a disutility of work of 0.01.
- For firms, price choices range from 0 to 2500 in units of 500; wage choices from 0 to 44 in units of 11.
- The 10 firms are split into two groups, receiving either 5000 or 10000 units of capital. Within these groups, firms receive a production exponent ranging from 0.2 to 0.8 in increments of 0.2. Thus each firm has a different production "technology".
- Firms invest 10% of their available budget (if positive) in each round to increase their capital stock.
- Government taxation choices range from 0 to 100% in units of 20%, for both income tax and corporate tax rates.
- The government can either value only consumers when calculating its welfare ("consumer-only") or value welfare of both consumers and firms ("total"), with firm welfare down-weighted by a factor of 0.0025 (to be commensurate with consumers).
- We set the minimum price at which firms are willing to export to be either 500 or 1000, and the quota for each firm's good to a variety of values: 10, 50, 100, or 1000.
- For consumers, consumption choices range from 0 to 10 units for each good and work choices from 0 to 1040 hours in increments of 260.

Figure 2: Key Simulation Implementations and Details Continued



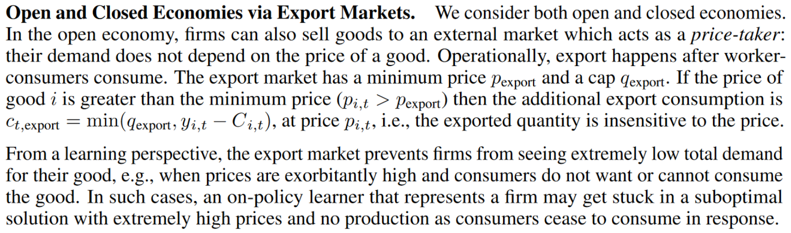
Agent Observations. The environment *state s* consists formally of all agent states and a general world state. Each agent observes can observe their own information and the global state:

$$s_{\text{global}} = \left(t, \{y_{i,t}\}_i, \{p_{i,t}\}_i, \{w_{i,t}\}_i, \{o_{i,t}\}_i\right). \tag{8}$$

Here $y_{i,t}$ is the available supply of good i, $p_{i,t}$ is the price, $w_{i,t}$ is the wage. The extra information $o_{i,t}$ includes whether good i was overdemanded at the previous timestep and tax information.

In addition, consumer-workers observe private information about their own state: $(B_{i,t}, \theta)$ A firm i also observes its private information: $(B_{i,t}, k_{i,t}, (0, \ldots, 1, \ldots, 0), \alpha)$, including a one-hot vector encoding its identity and its production function shape parameter α. The government only sees the global state.

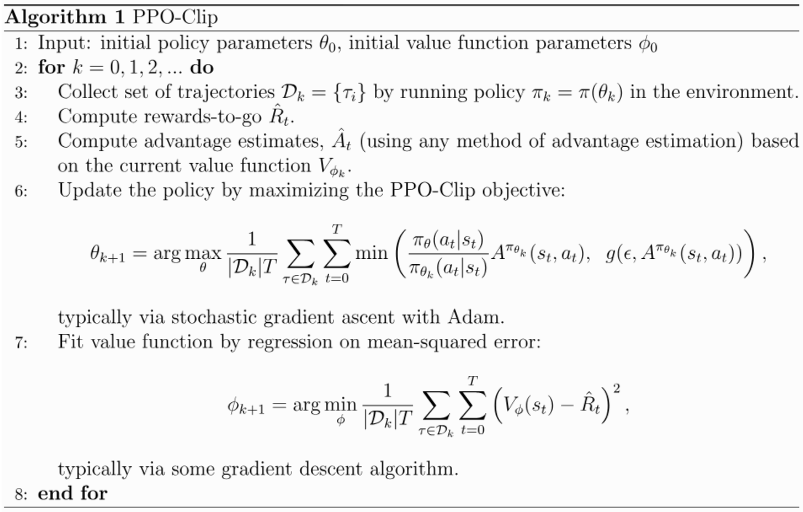
Figure 3: Key Simulation Implementations Notes



Open and Closed Economies via Export Markets. We consider both open and closed economies. In the open economy, firms can also sell goods to an external market which acts as a *price-taker*: their demand does not depend on the price of a good. Operationally, export happens after worker-consumers consume. The export market has a minimum price p_{export} and a cap q_{export}. If the price of good i is greater than the minimum price ($p_{i,t} > p_{\text{export}}$) then the additional export consumption is $c_{t,\text{export}} = \min(q_{\text{export}}, y_{i,t} - C_{i,t})$, at price $p_{i,t}$, i.e., the exported quantity is insensitive to the price.

From a learning perspective, the export market prevents firms from seeing extremely low total demand for their good, e.g., when prices are exorbitantly high and consumers do not want or cannot consume the good. In such cases, an on-policy learner that represents a firm may get stuck in a suboptimal solution with extremely high prices and no production as consumers cease to consume in response.

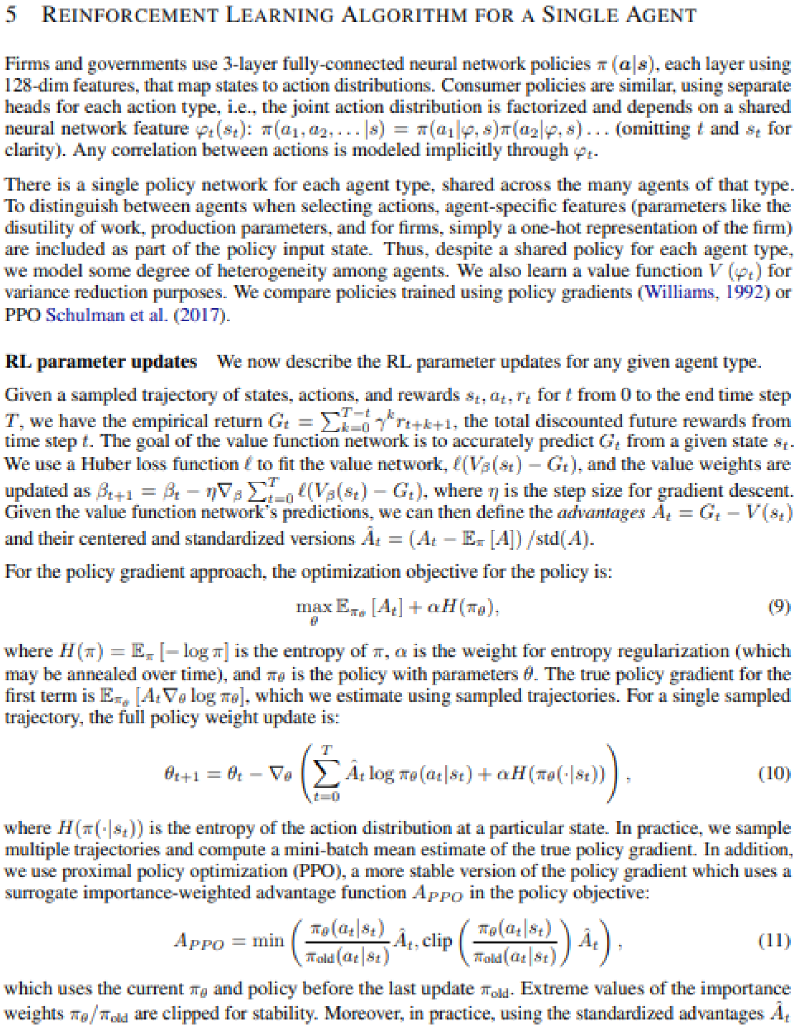
Figure 4: Open Economy Environment Description



Algorithm 1 PPO-Clip

```
Input: initial policy parameters $\theta_0$, initial value function parameters $\phi_0$
for $k = 0, 1, 2, ...$ do
    Collect set of trajectories $\mathcal{D}_k = \{\tau_i\}$ by running policy $\pi_k = \pi(\theta_k)$ in the environment.
    Compute rewards-to-go $\hat{R}_t$.
    Compute advantage estimates, $\hat{A}_t$ (using any method of advantage estimation) based
    on the current value function $V_{\phi_k}$.
    Update the policy by maximizing the PPO-Clip objective:
        $\theta_{k+1} = \arg\max_{\theta} \frac{1}{|\mathcal{D}_k| T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_k}(a_t|s_t)} A^{\pi_{\theta_k}}(s_t, a_t), \quad g(\epsilon, A^{\pi_{\theta_k}}(s_t, a_t))\right),$
    typically via stochastic gradient ascent with Adam.
    Fit value function by regression on mean-squared error:
        $\phi_{k+1} = \arg\min_{\phi} \frac{1}{|\mathcal{D}_k| T} \sum_{\tau \in \mathcal{D}_k} \sum_{t=0}^{T} \left(V_\phi(s_t) - \hat{R}_t\right)^2,$
    typically via some gradient descent algorithm.
end for
```

Figure 6: Algorithm for PPO



5 Reinforcement Learning Algorithm for a Single Agent

Firms and governments use 3-layer fully-connected neural network policies $\pi(\boldsymbol{a}|\boldsymbol{s})$, each layer using 128-dim features, that map states to action distributions. Consumer policies are similar, using separate heads for each action type, i.e., the joint action distribution is factorized and depends on a shared neural network feature $\varphi_t(s_t)$: $\pi(a_1, a_2, \ldots|s) = \pi(a_1|\varphi, s)\pi(a_2|\varphi, s)\ldots$ (omitting t and s_t for clarity). Any correlation between actions is modeled implicitly through φ_t.

There is a single policy network for each agent type, shared across the many agents of that type. To distinguish between agents when selecting actions, agent-specific features (parameters like the disutility of work, production parameters, and for firms, simply a one-hot representation of the firm) are included as part of the policy input state. Thus, despite a shared policy for each agent type, we model some degree of heterogeneity among agents. We also learn a value function $V(\varphi_t)$ for variance reduction purposes. We compare policies trained using policy gradients (Williams, 1992) or PPO Schulman et al. (2017).

RL parameter updates We now describe the RL parameter updates for any given agent type.

Given a sampled trajectory of states, actions, and rewards s_t, a_t, r_t for t from 0 to the end time step T, we have the empirical return $G_t = \sum_{k=0}^{T-t} \gamma^k r_{t+k+1}$, the total discounted future rewards from time step t. The goal of the value function network is to accurately predict G_t from a given state s_t. We use a Huber loss function ℓ to fit the value network, $\ell(V_\beta(s_t) - G_t)$, and the value weights are updated as $\beta_{t+1} = \beta_t - \eta\nabla_\beta \sum_{t=0}^{T} \ell(V_\beta(s_t) - G_t)$, where η is the step size for gradient descent. Given the value function network's predictions, we can then define the *advantages* $A_t = G_t - V(s_t)$ and their centered and standardized versions $\hat{A}_t = (A_t - \mathbb{E}_\pi[A])/\text{std}(A)$.

For the policy gradient approach, the optimization objective for the policy is:

$$\max_\theta \mathbb{E}_{\pi_\theta}[A_t] + \alpha H(\pi_\theta), \tag{9}$$

where $H(\pi) = \mathbb{E}_\pi[-\log \pi]$ is the entropy of π, α is the weight for entropy regularization (which may be annealed over time), and π_θ is the policy with parameters θ. The true policy gradient for the first term is $\mathbb{E}_{\pi_\theta}[A_t \nabla_\theta \log \pi_\theta]$, which we estimate using sampled trajectories. For a single sampled trajectory, the full policy weight update is:

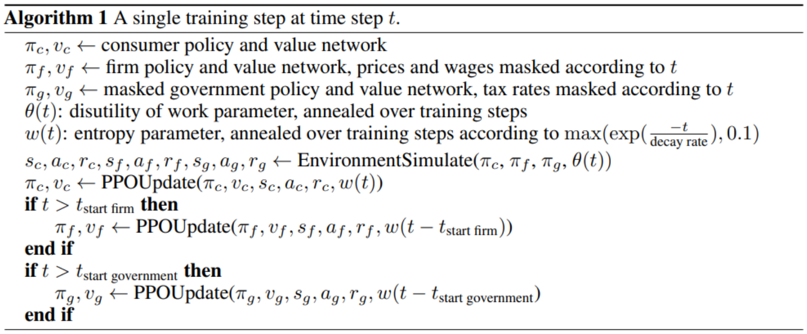
$$\theta_{t+1} = \theta_t - \nabla_\theta \left(\sum_{t=0}^{T} \hat{A}_t \log \pi_\theta(a_t|s_t) + \alpha H(\pi_\theta(\cdot|s_t)) \right), \tag{10}$$

where $H(\pi(\cdot|s_t))$ is the entropy of the action distribution at a particular state. In practice, we sample multiple trajectories and compute a mini-batch mean estimate of the true policy gradient. In addition, we use proximal policy optimization (PPO), a more stable version of the policy gradient which uses a surrogate importance-weighted advantage function A_{PPO} in the policy objective:

$$A_{PPO} = \min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\text{old}}(a_t|s_t)} \hat{A}_t, \text{clip}\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\text{old}}(a_t|s_t)} \right) \hat{A}_t \right), \tag{11}$$

which uses the current π_θ and policy before the last update π_{old}. Extreme values of the importance weights $\pi_\theta/\pi_{\text{old}}$ are clipped for stability. Moreover, in practice, using the standardized advantages $\hat{A}_t$

Figure 7: PPO Updates in Economy



Algorithm 1 A single training step at time step t.

```
$\pi_c, v_c \leftarrow$ consumer policy and value network
$\pi_f, v_f \leftarrow$ firm policy and value network, prices and wages masked according to $t$
$\pi_g, v_g \leftarrow$ masked government policy and value network, tax rates masked according to $t$
$\theta(t)$: disutility of work parameter, annealed over training steps
$w(t)$: entropy parameter, annealed over training steps according to $\max(\exp(\frac{-t}{\text{decay rate}}), 0.1)$
$s_c, a_c, r_c, s_f, a_f, r_f, s_g, a_g, r_g \leftarrow$ EnvironmentSimulate($\pi_c$, $\pi_f$, $\pi_g$, $\theta(t)$)
$\pi_c, v_c \leftarrow$ PPOUpdate($\pi_c, v_c, s_c, a_c, r_c, w(t)$)
if $t > t_{\text{start firm}}$ then
    $\pi_f, v_f \leftarrow$ PPOUpdate($\pi_f, v_f, s_f, a_f, r_f, w(t - t_{\text{start firm}})$)
end if
if $t > t_{\text{start government}}$ then
    $\pi_g, v_g \leftarrow$ PPOUpdate($\pi_g, v_g, s_g, a_g, r_g, w(t - t_{\text{start government}})$)
end if
```

Figure 8: Algorithm for PPO in Economy

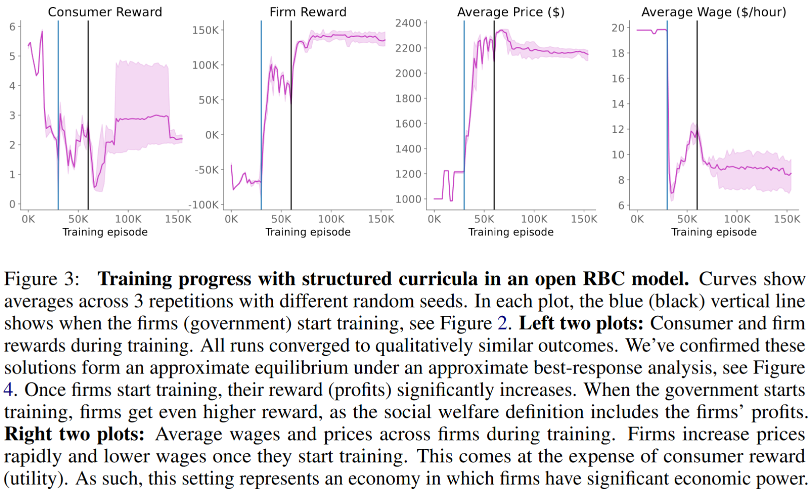


Figure 3: **Training progress with structured curricula in an open RBC model.** Curves show averages across 3 repetitions with different random seeds. In each plot, the blue (black) vertical line shows when the firms (government) start training, see Figure 2. **Left two plots:** Consumer and firm rewards during training. All runs converged to qualitatively similar outcomes. We've confirmed these solutions form an approximate equilibrium under an approximate best-response analysis, see Figure 4. Once firms start training, their reward (profits) significantly increases. When the government starts training, firms get even higher reward, as the social welfare definition includes the firms' profits. **Right two plots:** Average wages and prices across firms during training. Firms increase prices rapidly and lower wages once they start training. This comes at the expense of consumer reward (utility). As such, this setting represents an economy in which firms have significant economic power.

Figure 16: Original Environment Results

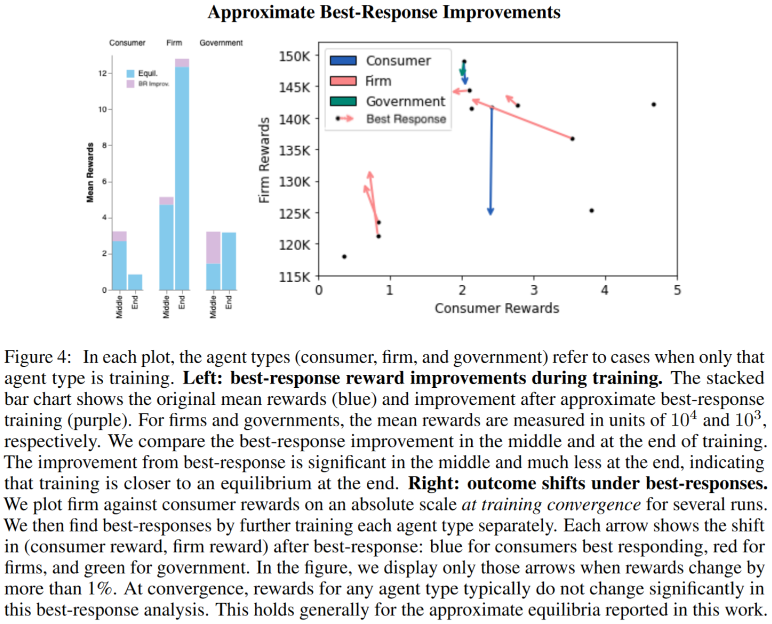


Figure 4: In each plot, the agent types (consumer, firm, and government) refer to cases when only that agent type is training. **Left: best-response reward improvements during training.** The stacked bar chart shows the original mean rewards (blue) and improvement after approximate best-response training (purple). For firms and governments, the mean rewards are measured in units of 10^4 and 10^3, respectively. We compare the best-response improvement in the middle and at the end of training. The improvement from best-response is significant in the middle and much less at the end, indicating that training is closer to an equilibrium at the end. **Right: outcome shifts under best-responses.** We plot firm against consumer rewards on an absolute scale *at training convergence* for several runs. We then find best-responses by further training each agent type separately. Each arrow shows the shift in (consumer reward, firm reward) after best-response: blue for consumers best responding, red for firms, and green for government. In the figure, we display only those arrows when rewards change by more than 1%. At convergence, rewards for any agent type typically do not change significantly in this best-response analysis. This holds generally for the approximate equilibria reported in this work.

Figure 17: Original Equilibrium Analysis