Aalto University
School of Electrical
Engineering

Master's Programme in International Design Business Management

Reinforcement Learning for Firm Behavior in a Macroeconomic Agent-Based Model

Proximal Policy Optimization, Profit Optimization, and Systemic Outcomes

Hieu Le

Master's Thesis
2026

Aalto University
School of Electrical
Engineering

Author Hieu Le

Title Reinforcement Learning for Firm Behavior in a Macroeconomic Agent-Based Model — Proximal Policy Optimization, Profit Optimization, and Systemic Outcomes

Degree programme International Design Business Management

Major International Design Business Management

Supervisor and advisor Prof. Ahti Salo

Date 29 July 2026 **Number of pages** 89 **Language** English

Abstract

This thesis investigates the behavior and systemic effects of a profit-seeking firm in a macroeconomic agent-based model by extending the Lengnick baseline model with a recurrent reinforcement learning policy trained using Proximal Policy Optimization. Based only on local observations, the policy determines the firm's wage, price, and workforce decisions. Its performance is compared with that of the baseline heuristic, and its systemic effects are evaluated under adoption levels of 1%, 50%, and 100%, where adoption refers to the share of firms controlled by the learned policy.

The learned firm develops a scale-expansion strategy characterized by lower relative prices, higher relative wages, and workforce expansion. It survives the 100-year evaluation horizon in 15 of the 16 runs in which it is economically active at the start of evaluation, compared with 2 of 320 matched baseline runs, and earns more than twice the baseline heuristic's mean monthly profit over the first ten years.

At the system level, all adoption scenarios reduce unemployment and improve the fulfillment of effective household demand relative to the baseline, although the effects are nonlinear. At 50% adoption, mean unemployment falls from 0.54% to 0.07%, while the mean household unsatisfied demand ratio falls to about one fifth of its baseline level. Under full adoption, unemployment is almost eliminated, but inventories rise substantially and unsatisfied demand increases relative to 50% adoption, indicating persistent excess supply.

The findings demonstrate that reinforcement learning can identify profitable behavioral strategies and reveal unexpected systemic effects in macroeconomic agent-based models, while showing that behavior that is successful for an individual firm does not necessarily produce proportionate improvements when widely adopted.

Keywords agent-based modeling, reinforcement learning, firm behavior, profit optimization, macroeconomic simulation, computational economics

Preface

I would like to express my sincere gratitude to my supervisor, Professor Ahti Salo, for his guidance and feedback throughout this work. I very much appreciate both the degree of creative freedom he gave me and the structure that helped me stay focused.

My studies in the Master's Programme in International Design Business Management have played a central role in shaping both the topic and approach of this thesis. I am grateful to my teachers and classmates for instilling in me an interdisciplinary perspective and broadening my horizons on subjects beyond my own area of expertise.

Finally, I would like to thank my family and friends. Their love and support have been indispensable throughout this journey; without them, none of this would have been possible.

Helsinki, 29 July 2026

Hieu Le

Contents

Symbols and abbreviations

Symbols

Reinforcement learning notation

t	Discrete time step
T	Terminal time step of an episode
K	Final state index of a sampled rollout segment
$\mathcal{S}, \mathcal{A}, \mathcal{R}$	State, action and reward spaces of a Markov decision process
S_t, A_t, R_{t+1}	State and action at time step t, and the resulting reward
$p(s', r \mid s, a)$	Probability distribution governing the environment dynamics
G	Discounted return over an episode
G_t	Discounted return from time step t onward
γ	Discount rate
π	Decision-making policy
π_θ	Stochastic policy parameterized by θ
$\theta, \theta_{\text{old}}$	Candidate and data-generating policy parameters
$J(\theta)$	Scalar performance measure for the policy π_θ
$v_\pi(s), q_\pi(s, a)$	State-value and action-value functions under policy π
$v_*(s), q_*(s, a)$	Optimal state-value and action-value functions
$\text{Adv}_\pi(s, a)$	Advantage function under policy π
$\widehat{\text{Adv}}_t$	Estimated advantage of the sampled action at time t
$\hat{v}(s)$	State-value estimate produced by the critic
V_t^{targ}	Fixed target used to train the critic
δ_t	One-step temporal-difference residual
λ_{GAE}	Weighting parameter used by Generalized Advantage Estimation
$\widehat{\mathbb{E}}_t[\cdot]$	Empirical average over sampled time steps
$r_t(\theta)$	Likelihood ratio between the candidate and data-generating policies
ϵ	PPO clipping hyperparameter
L^{PG}	Empirical policy-gradient objective
L^{CPI}	Conservative policy iteration surrogate objective
L^{CLIP}	Clipped surrogate objective
L^{V}	Value-function loss
L^{PPO}	Overall PPO objective
c_V, c_H	Coefficients of the value-function loss and entropy bonus
$\mathcal{H}[\pi_\theta](S_t)$	Entropy of the policy's action distribution in state S_t

Economic model notation

h, f	Household and firm indices
H, F	Numbers of households and firms
A_h	Set of suppliers connected to household h
B_f	Set of households employed by firm f
m_h, m_f	Liquidity of household h and firm f
m_f^{buf}	Liquidity buffer of firm f
ω_h	Reservation wage of household h
i_f	Inventory of firm f
p_f, w_f	Goods price and monthly wage set by firm f
v_f, s_f	Vacancy indicator and consecutive fully staffed months of firm f
P_h^I	Mean goods price among the suppliers of household h
d_f	Demand encountered by firm f during the previous month
c_h^r	Planned monthly real consumption of household h
d_h	Daily consumption target of household h
$\tilde{d}_h^{(k)}$	Effective demand of household h in purchase attempt k
$q_h^{(k)}$	Quantity purchased by household h in purchase attempt k
$u_{h,f}$	Unsatisfied demand experienced by household h at firm f
$\underline{i}_f, \overline{i}_f$	Lower and upper inventory thresholds of firm f
mc_f	Marginal cost per unit of firm f
$\underline{p}_f, \overline{p}_f$	Lower and upper goods-price thresholds of firm f
M_H	Aggregate household liquidity before profit distribution
n_A	Number of supplier connections per household
α_{elas}	Elasticity of planned real consumption with respect to real wealth
ξ	Relative price threshold for replacing a supplier
$\psi_{\text{price}}, \psi_{\text{quant}}$	Probabilities of price-based and demand-based supplier search
β	Job-search attempts of an unemployed household
π_{emp}	Job-search probability of a satisfied employed household
γ_{month}	Fully staffed months required before a wage reduction
δ, ϑ	Upper bounds of wage and goods-price adjustment rates
$\underline{\phi}, \overline{\phi}$	Lower and upper inventory-threshold multipliers
$\underline{\varphi}, \overline{\varphi}$	Lower and upper price-threshold multipliers relative to marginal cost
θ_{calvo}	Probability of implementing a newly determined goods price
λ	Units of consumption goods produced by one worker per day
χ	Fraction of monthly labor costs retained as a liquidity buffer

Learned firm extension

o_t	Observation vector received by the learned firm at month t
a_t	Mixed action vector selected by the learned firm at month t
a_t^w	Wage adjustment
a_t^p	Goods price adjustment
a_t^{hr}	Workforce action
$R_{f,t}^{\mathrm{sales}}$	Realized sales revenue of firm f during month t
$C_{f,t}^{\mathrm{labor}}$	Labor costs of firm f during month t
$\Pi_{f,t}^{\mathrm{nom}}$	Nominal profit of firm f during month t
CPI_t	Consumer price index in month t
$\mathcal{K}_{h,f,t}$	Purchase attempts by household h at firm f in month t
κ	Profit-scaling hyperparameter in the reward function
ι	Survival-bonus hyperparameter in the reward function
I_{t+1}^{alive}	Indicator that the learned firm remains viable after month t

Abbreviations

ABM	Agent-based model
ACE	Agent-based computational economics
CPI	Consumer price index
GAE	Generalized Advantage Estimation
GPI	Generalized Policy Iteration
HHI	Herfindahl–Hirschman index
HR	Human resources
LBM	Lengnick baseline model
LSTM	Long short-term memory
MDP	Markov decision process
MPS	Metal Performance Shaders
PPO	Proximal Policy Optimization
RL	Reinforcement learning

1 Introduction

In a market economy, one of the primary goals of firms is to generate profit by satisfying customer demand efficiently while remaining competitive. Through their decisions regarding prices, wages, production and employment, however, firms influence not only their own profitability, but also the allocation of labor, the availability of goods and the distribution of liquidity throughout the economy. Understanding firm behavior and the mechanisms through which firms pursue profit is therefore important for understanding the macroeconomic patterns that emerge from their interactions.

Agent-based models (ABMs) simulate individual autonomous agents in order to study how agent-level interactions produce system-level outcomes that may not be easy to predict [22, Ch. 8]. As a tool for investigating macroeconomic phenomena, this bottom-up approach differs from many conventional macroeconomic models that rely on equilibrium assumptions: it allows heterogeneity, bounded rationality, market frictions and non-equilibrium dynamics to be represented explicitly [6]. In exchange for this flexibility, however, designing and justifying agent behavior can be challenging, particularly in behaviorally rich models containing many interacting decisions. While hand-crafted behavioral rules can make a model transparent and tractable, and may reproduce relevant stylized facts, they do not necessarily represent effective behavior from the standpoint of the individual agent. Recent research has therefore examined reinforcement learning as one possible method for constructing adaptive behavior within economic models [3, 7].

Reinforcement learning (RL) is an area of machine learning concerned with optimizing an agent's sequential decision-making process in order to maximize a defined objective [25]. In theory, this framework can be particularly well suited for agent-based modeling because the consequences of an agent's actions can develop over time, depending on its interactions with other agents. Instead of specifying an explicit behavioral rule for every possible state, modelers can train an agent within the simulated environment using a reward function that represents its objective. In economics, this reward may correspond to an agent's utility, profit or another measure of performance. RL can therefore serve as an experimental tool for determining whether alternative strategies exist within a model and for investigating which model mechanisms allow those strategies to succeed. At the same time, because learned policies may be less transparent than hand-crafted heuristics, additional effort is required to interpret and analyze their behavior and system-level consequences.

This thesis investigates the application of RL to firm behavior in the Lengnick

baseline model (LBM), a stylized and parsimonious macroeconomic ABM [16]. Previous studies have demonstrated that RL can be used to solve economic decision problems and introduce adaptive agents into macroeconomic simulations [3, 5, 7]. Building on this research, the thesis focuses on the detailed behavior of a learned firm, the systemic outcomes associated with its introduction and the mechanisms linking firm-level behavior to system-level outcomes. It also examines the deployment of the learned strategy at different adoption levels. Deploying the same policy to an increasing share of the firm population makes it possible to study how the performance and consequences of an individually successful strategy change as that strategy becomes more widespread.

Specifically, this thesis examines whether RL can produce a firm policy that outperforms the baseline heuristic in generating nominal profit within the context of the LBM. It then investigates the behavioral strategy through which the learned firm achieves its performance and analyzes how the adoption of the learned policy affects market structure and macroeconomic outcomes.

The thesis accomplishes these objectives by replacing a baseline firm in the LBM with a learned firm controlled by a recurrent policy trained using Proximal Policy Optimization (PPO) [24], an RL algorithm designed to promote stable policy updates. PPO is suitable for the firm decision problem because it can be applied to both continuous and discrete decisions. The policy architecture accommodates this mixed action space, while its recurrent component enables the policy to retain information from previous observations and actions. The policy is trained in an economy consisting of one learned firm and 99 baseline firms, after which it is evaluated and analyzed across held-out model seeds. The learned policy's systemic effects are examined under four adoption scenarios: no learned firms (0%), one learned firm (1%), 50 learned firms (50%) and 100 learned firms (100%). Notably, the 50% and 100% adoption scenarios represent scaled deployments of a policy trained in the setting with one learned firm, and no separate multi-agent optimization was used in those scenarios.

The first contribution of this thesis is to demonstrate how recurrent PPO can be applied to firm behavior within a macroeconomic ABM. This includes formulating the learning problem, designing the environment, specifying the reward function, implementing and configuring a recurrent PPO architecture for the firm's mixed action space and evaluating the resulting policy. Second, the thesis provides a detailed interpretation of the learned policy and a mechanism-oriented analysis of the resulting macroeconomic outcomes. Rather than evaluating the policy solely through performance metrics, the analysis examines how its wage, price and workforce

decisions develop into an economically interpretable strategy. Third, by deploying the same learned policy at different adoption levels, the thesis examines how the performance and systemic consequences of an individually successful strategy change as that strategy becomes more widespread.

The remainder of the thesis is structured as follows. Section 2 reviews the relevant background on agent-based modeling and RL. Section 3 presents the LBM, the RL extension and the methodology for training and evaluation. Section 4 reports the firm-level results and the macroeconomic outcomes under different adoption scenarios. Section 5 interprets the learned strategy and the emergent system-level mechanisms, discusses the limitations and proposes directions for future research. Section 6 concludes the thesis.

2 Background

2.1 Reinforcement Learning (RL)

2.1.1 Core concepts and terminology

At a high level, reinforcement learning is the study of learning how to act according to a specified goal by exploring and interacting with the environment, without explicit instructions on the optimal strategy [25]. The specification of the goal is embedded in the evaluative feedback that the environment provides to the learner after every action taken, and the learner must utilize this feedback to improve its decision-making process. A key feature of reinforcement learning is that, in order to achieve the specified goal, the learner must take into account all of the feedback signals over a course of actions, not just the immediate feedback signal of the current action: Selecting actions with short-term benefits might affect the environment in ways that could obstruct the long-term target of achieving the goal. This emphasis on delayed outcomes and future planning, along with its general-purpose nature, makes RL methods a useful tool to investigate and solve a wide range of sequential decision-making problems.

Markov Decision Processes Markov Decision Processes (MDPs) are mathematical formalizations of the problems RL aims to solve. An MDP involves a decision-making learner, termed the *agent*, interacting sequentially with an *environment*, which encompasses everything defined in the problem except the agent. The goal of the agent is to maximize the *cumulative reward*, which consists of the individual *rewards* given by the environment as evaluative feedback after the agent has taken an *action*. In addition to the rewards, the basis for the agent's decision-making also includes the environment's *states*, which represent all the information about the environment that the agent could use to make decisions.

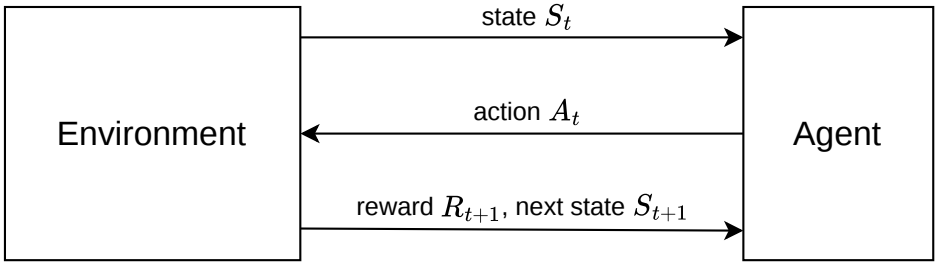

Figure 1: Agent–environment interaction in RL. At time step t, the agent observes state S_t, selects action A_t, and receives reward R_{t+1} and next state S_{t+1} from the environment. Adapted from [25, p. 48].

The sequentiality of the problem is expressed in the notion of time steps, $t = 0, 1, 2, ...$ The order of operation in a specific time step t is as follows:

1. The agent observes the environment's current state $S_t \in \mathcal{S}$
2. The agent takes an action $A_t \in \mathcal{A}$
3. The agent receives a reward $R_{t+1} \in \mathcal{R}$
4. The state of the environment becomes $S_{t+1} \in \mathcal{S}$ and the operation repeats for time step $t + 1$

The set of states $\mathcal{S}$ and the set of actions $\mathcal{A}$ are respectively called the *state space* and the *action space* of an MDP. The set of rewards $\mathcal{R} \subset \mathbb{R}$ consists of immediate real-value rewards given by the environment after the agent takes an action and enters a new state of the environment. This thesis follows the notation used in Sutton and Barto [25, Ch. 3], including the notation of reward R_{t+1} following action A_t, to emphasize that the agent receives the information of the reward and the next state at the same time (although the R_t convention is also used in other academic texts).

The *dynamics* of an MDP, or how an MDP works, are described through a *dynamics function* $p : \mathcal{S} \times \mathcal{R} \times \mathcal{S} \times \mathcal{A} \to [0, 1]$, defined as

$$p(s', r \mid s, a) \doteq \Pr\{S_{t+1} = s', R_{t+1} = r \mid S_t = s, A_t = a\}. \tag{1}$$

The function p determines the probability that the agent taking action a in state s at time step t will arrive in state s' with reward r at time step $t + 1$. The probability distribution specified by the function p is a complete description of the dynamics of an MDP, as in the following equation. For readability, the formula is presented

for discrete spaces using summation; in the case of continuous spaces, sums become integrals and p denotes a density:

$$\sum_{s' \in \mathcal{S}} \sum_{r \in \mathcal{R}} p(s', r \mid s, a) = 1, \quad \forall s \in \mathcal{S},\ \forall a \in \mathcal{A}.$$

An implication of the completeness of description is that what happens in an MDP at time step $t + 1$, the values of S_{t+1} and R_{t+1}, depend only on the last state S_t and action A_t, and not on the states and actions before t. If a state contains all the relevant information, independent of the MDP's history, to determine the immediate next state and reward given an action, it is considered to have the *Markov property*. A majority of standard RL methods require the *Markov assumption*, which assumes that the Markov property holds for every state of the MDP.

In practice, a wide range of sequential decision-making problems can be reduced to MDPs, collections of $(\mathcal{S}, \mathcal{A}, \mathcal{R}, p)$, even though MDP as a framework is a considerable simplification compared to the complexity of real-world problems. The idea of using numerical reward signals to represent goals is central to MDP and RL, and careful design of these reward signals is essential to producing agents that solve MDPs in the intended way. Without satisfactory reward design for an MDP, RL methods produce agents that do not solve the intended problem that the MDP represents, or solve the problem in unintended ways.

Objectives and returns The decision-making task that the agent has to perform can either be *episodic* or *continuing*, respectively when the agent-environment interaction sequence has clear beginning and end (playing a chess game as an example), or goes on continuously without end (stock trading as an example).

In the case of episodic tasks, an interaction sequence going from beginning to end is called an *episode*, of which the final end state is called the *terminal state* (or *done state* sometimes in technical implementation) S_T, with T denoting the time step that the episode ends on. An agent's informal objective of maximizing the cumulative reward can be expressed formally as the maximization of the expected *discounted return*, where the discounted return G of an episode is

$$G \doteq \sum_{k=1}^{T} \gamma^{k-1} R_k, \tag{2}$$

where $0 \leq \gamma \leq 1$ is called the *discount rate* (or *discount factor*). Note that the notation of state and reward is respectively S_t and R_{t+1} within a specific time step t. The

terminal state is S_T, but the last decision time is $t = T - 1$: at time $t = T - 1$ the agent observes S_{T-1}, chooses A_{T-1}, receives the final reward R_T, and transitions to the terminal state S_T. Thus an episode contains the rewards $R_1, \ldots, R_T$, which is why the discounted return in Equation (2) sums over $k = 1, \ldots, T$.

The intuition behind the discounting concept is that in most real-world decision-making scenarios, there is a preference for immediate rewards to future ones. A straightforward example is the case of money, where receiving a certain amount of money now is more beneficial than receiving the same amount of money in the future (the time value of money in the economics literature). In addition, discounting accounts for the risk and uncertainty in stochastic environments, where the further out the reward is in the future, the less certain it is to be received. The value of γ varies by the nature of the task and the preference of the solution designer. A discount rate closer to 0 implies a desire for a more myopic agent that aims to maximize immediate rewards, whereas a discount rate closer to 1 means a farsighted agent is preferred.

Another benefit of having a discount rate is in the formalization of returns for continuing tasks, where effectively $T = \infty$. With $\gamma < 1$, the definition of G in Equation (2) can also be applied to the case of continuing tasks; the value of G is finite as long as R_k is bounded, even if the task goes on forever.

While G denotes the episodic return, G_t generalizes this to any time step t and denotes the cumulative discounted return from time step t onward, after taking action A_t:

$$G_t \doteq \sum_{k=t+1}^{T} \gamma^{k-t-1} R_k.$$

This formulation is useful in reducing the agent's overall objective of maximizing the expected discounted return $\mathbb{E}[G]$ to the individual objective in each time step t: choosing an action A_t so as to maximize the expected discounted return $\mathbb{E}[G_t]$ at time step t, where $\mathbb{E}[\cdot]$ denotes the expected value of the random variable inside the brackets.

Policies, value functions and optimality The decision-making behavior of an agent, or how an agent chooses an action, is described by a *policy* $\pi : \mathcal{A} \times \mathcal{S} \to [0, 1]$, defined as

$$\pi(a \mid s) \doteq \Pr(A_t = a \mid S_t = s).$$

The policy π determines the probability of the agent choosing action A_t when in state S_t. For every state in the state space, the policy π specifies a probability distribution over the action space, as in the following equation. For readability, the formula is presented for discrete spaces using summation; in the case of continuous spaces, sums become integrals and π denotes a density for continuous actions:

$$\sum_{a \in \mathcal{A}} \pi(a \mid s) = 1, \quad \forall s \in \mathcal{S}.$$

The notion of policy is required to quantify the performance of the agent, not only because it is the formal representation of how the agent acts, but also because the expected discounted return (the objective that the agent wants to maximize) must be calculated with respect to a specific decision-making behavior: the rewards the agent receives depend on what actions are chosen. The *state-value function* $v_\pi(s)$ is one formalization of the expected discounted return with regard to a specific policy π, defined as

$$v_\pi(s) \doteq \mathbb{E}_\pi\big[G_t \mid S_t = s\big],$$

where $\mathbb{E}_\pi[\cdot]$ denotes the expected value of the random variable inside the brackets, provided that the agent follows policy π. The function $v_\pi(s)$ is the discounted return the agent can expect to receive when it starts in state s and chooses actions according to policy π. The time step t in the definition of $v_\pi(s)$ can be any value, as its role in the equation is to provide a bridge between the discounted return G_t and the starting state S_t. In RL algorithms, it is usually more useful to know how desirable it is for the agent to be in certain states than for the agent to be in certain time steps.

Similar to the state-value function $v_\pi(s)$, another value function is the *action-value function* $q_\pi(s, a)$, defined as

$$q_\pi(s, a) \doteq \mathbb{E}_\pi\big[G_t \mid S_t = s, A_t = a\big].$$

The function $q_\pi(s, a)$ is the discounted return the agent can expect to receive when it takes action a in state s and subsequently chooses actions according to policy π. The value functions $v_\pi(s)$ and $q_\pi(s, a)$ provide a way to evaluate and compare different policies. A policy π is better than or equal to a policy π' if and only if the expected return following π is better than or equal to the expected return following π' across the state space:

$$\pi \geq \pi' \iff v_\pi(s) \geq v_{\pi'}(s), \quad \forall s \in \mathcal{S}.$$

An *optimal policy* π_* is a policy that is better than or equal to all the other policies. Within a specific MDP, there may be multiple optimal policies, but all of them will have the same *optimal state-value function* $v_*(s)$, defined as

$$v_*(s) \doteq \max_\pi v_\pi(s), \quad \forall s \in \mathcal{S}.$$

Similarly, the optimal policies have the same *optimal action-value function* q_*, defined as

$$q_*(s,a) \doteq \max_\pi q_\pi(s,a), \quad \forall s \in \mathcal{S},\ \forall a \in \mathcal{A}.$$

Considering the objective of an agent (choosing A_t so as to maximize $\mathbb{E}[G_t]$), the ultimate goal of reinforcement learning is to find a way to reach an optimal policy π_* that accomplishes that objective. An example approach to achieve that goal is to use the optimal value functions $v_*(s)$ and $q_*(s,a)$. Given the optimal value function, it is relatively straightforward to determine an optimal policy. In the case that $v_*(s)$ is provided, the agent only needs to search one-step forward to find the action that leads to the state with the highest optimal state value. Because $v_*(s)$ has already accounted for all future rewards and optimal actions, the agent does not need to look further than the next action it is considering. Similarly, having $q_*(s,a)$ achieves the same optimal policy.

In theory, it is possible to calculate $v_*(s)$ and $q_*(s,a)$ (that would lead to π_*) for certain MDPs given a complete and accurate model of the environment. However, it is often intractable to compute an optimal policy for MDPs representing real-world problems, due to the complexity of the dynamics functions or technical constraints, computation-wise or memory-wise. Therefore, optimal (or sufficiently near-optimal) policies and value functions can only be approximated for such problems.

2.1.2 Overview of RL methods and algorithms

A consequence of RL's wide applicability in concept is its wide breadth of methods and algorithms. Different problem specifications lead to different MDP abstractions, which subsequently lead to specific and focused methods being developed to address specific problems arising in those scenarios. An exhaustive list of available RL methods and algorithms is beyond the scope of this thesis, but key themes of RL methods can be

identified and discussed, with the goal of providing enough background to explain and discuss the specific methods used in this thesis.

Within this thesis, an algorithm is considered a precise sequence of steps for solving a specific and narrow computational problem, whereas a method (or approach) is a more general and systemic procedure used to solve a broader problem. In a sense, a method is larger than an algorithm and a method can contain multiple algorithms, including the interactions between the component algorithms. Lastly, a technique is a flexible and modular tool or "trick" applied in practice to improve the performance of other methods and algorithms.

Exploration and exploitation One of the unique challenges in RL is the trade-off between *exploration* and *exploitation*. In order to maximize the cumulative reward, the agent has to exploit its experience, choosing actions that have proven to be effective from past interactions with the environment. However, discovering such optimal actions requires the agent to explore, trying actions it has not chosen before but that might yield better rewards in the future. It is not possible for an RL method to include only exploration or only exploitation if the goal is to obtain an optimal policy in a reasonable timeframe. Insufficient exploration can trap the agent in a suboptimal policy, while excessive exploration prolongs the time it takes to reach optimality (or near-optimality). The appropriate balance between exploration and exploitation (and the balancing mechanisms) for a specific problem depends on the underlying MDP, the amount of available data (the agent-environment interactions) and the time constraints of the solution.

Prediction and control problems A broad categorization of RL algorithms separates them into *prediction algorithms* and *control algorithms*. In a prediction problem (also referred to as policy evaluation problem), the objective of the algorithm is to evaluate a fixed policy π, that is, to estimate its state-value function v_π or action-value function q_π as accurately as possible. In contrast, a control problem (also referred to as policy improvement problem) is concerned with finding a policy that is optimal or near-optimal, typically by iteratively constructing policies that are better than the current one until no further improvement can be found.

Often, control algorithms make use of value estimates produced by prediction subroutines to improve the current policy, by acting *greedily* (or near-greedily with room for exploration) with respect to the current value estimates—that is, by choosing actions that yield the highest discounted return according to those estimates. Many

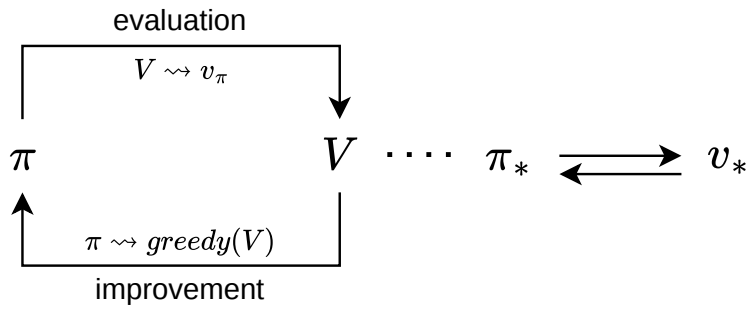

Figure 2: Generalized policy iteration: policy evaluation improves the value function under the current policy, and policy improvement updates the policy toward greediness with respect to the value function. Adapted from [25, p. 87].

RL methods can therefore be understood as repeatedly alternating between solving a prediction subproblem and performing a control step. *Generalized policy iteration* (GPI) is the term used for the general pattern in which policy-evaluation and policy-improvement processes interact continually until they stabilize, ideally at or near an optimal policy and value function. Figure 2 illustrates this continual interaction between policy evaluation and policy improvement.

While most RL methods can be framed as GPI, some methods are purely control-oriented: they update the policy directly on the basis of observed returns or performance signals, without maintaining an explicit estimate of the current policy's value function. In such methods, the emphasis is on adjusting behavior to improve performance, rather than on separately solving a prediction subproblem.

Tabular and approximate methods As briefly mentioned at the end of Section 2.1.1, optimal value functions and policies can be, in principle, calculated exactly given certain properties of the MDP. One of the requirements for this calculation to be possible is the tabular representation of value functions. In tabular settings, the sizes of the state spaces and action spaces of the MDPs are small enough that the value functions can be represented explicitly per state (or state-action pair) as rows in tables (or as arrays). Given a complete model of the environment, an MDP in a tabular setting can be solved using dynamic programming, an optimization method from which RL borrows many key ideas.

Many real-world tasks, however, require MDP abstractions that have too complicated dynamics for tabular methods to be useful. A high-dimensional and large state space can produce an enormous number of possible states, such that an optimal

policy or the optimal value function cannot be found even with infinite time and data [25, p. 195]. In those complex scenarios, approximate solution methods are required given the limits of time and computational resources. Approximate methods allow agents to generalize across similar states, whereas tabular methods treat every state independently. Generalization is especially important in cases where each unique state is visited only a few times, making learning impossible using tabular methods.

Value-based and policy-based algorithms The distinction between *value-based* and *policy-based* methods is closely related to the division between prediction and control problems discussed above. In value-based methods, the central objective is the learning of the optimal value function, from which an optimal policy is derived as a result of acting greedily (or near-greedily) with respect to the value estimates. In contrast, the principal target of policy-based methods (also referred to as policy-search methods) is to directly learn a parameterized policy that can select actions without consulting an explicit value function, often by performing gradient ascent (or related optimization) steps on the policy parameters based on the gradient of some scalar performance measure.

Briefly mentioned above as purely control-oriented methods, policy-based methods have some advantages over value-based methods. The direct parameterization of behavior makes it straightforward to represent stochastic policies, which can be advantageous for exploration and for problems in which optimal behavior is inherently stochastic [25, p. 323]. In addition, for some problems, estimating the optimal policy in a parameterized form is simply easier than estimating the value function [25, p. 324], especially in the case of high-dimensional and continuous action spaces [14, p. 20], where approximating the value function is computationally demanding and prone to instability.

Regardless of the problem setting, however, policy-based methods have a few key weaknesses. Because of the way the gradient is estimated, policy-based methods can suffer from high-variance gradient estimates and, as a consequence, slow learning [25, pp. 329, 331], which can make it challenging to apply them to problems where only a small amount of sample data is available. Moreover, because gradient-based searches often only consider local neighborhoods of the parameter space, policy-based methods are more susceptible to convergence to local optima; global optimality is not guaranteed [14, Sec. 2.3]. In practice, these drawbacks can translate into high sensitivity to hyperparameters and the need for large amounts of interaction data.

To address some of the disadvantages of policy-based methods, *actor-critic*

methods were developed to leverage the strengths of value-based methods. Similar to policy-based methods, actor-critic methods maintain explicit parameterized policies (the actor) but are additionally coupled with (often also parameterized) value functions (the critic). Whereas the value function determines the policy in value-based methods, in actor-critic methods the critic only suggests a direction that the actor can follow. This architecture allows the critic to provide lower-variance learning signals for the actor, while the actor retains the flexibility of direct policy parameterization and the ability to handle continuous action spaces [25, p. 332, 14, Sec. 2.3]. As a trade-off, actor-critic methods are typically more computationally intensive and can introduce additional bias in the policy gradient due to approximate, bootstrapped value estimates; however, in practice these are often favorable trade-offs. The methods employed in this thesis are of the actor-critic class. Details are discussed at greater depth in Section 2.4.

Model-based and model-free algorithms As noted in Section 2.1.1, exact computation of value functions relies on a complete and accurate model of the environment. The availability of such a model is what distinguishes *model-based* from *model-free* methods. In model-based methods, the agent either has access to, or attempts to learn, an explicit model of the environment, generally in the form of the dynamics function p defined in Equation (1). Using the model, the agent can simulate the consequences of different actions in the environment and learn from those simulated experiences—in other words, the agent can *plan* ahead, which is advantageous in problems where real-world interaction data are limited or costly to obtain [25, p. 162, 19, Sec. 7.1]. However, learning a sufficiently accurate model of an environment with complex dynamics can be technically challenging and computationally expensive, and even small modeling errors can lead to substantial bias in the learned policies [19, Sec. 7.7]. In those scenarios, model-free methods are often preferred due to their ease of implementation and practicality, at the cost of lower sample efficiency compared to model-based methods.

On-policy and off-policy algorithms RL methods can be grouped by the source of the data they learn from through interaction with the environment. *On-policy* and *off-policy* methods differ in whether the interaction data come from the same policy that the agent is learning or from a different one. In off-policy methods, two policies are maintained simultaneously—a *behavior policy* used to generate (typically more exploratory) actions and a *target policy* used for optimization. Off-policy learning as

a concept is general and compelling, as it encompasses on-policy methods (special cases where target and behavior policies are the same) [25, p. 103]. This decoupling, in principle, suggests greater data reusability and sample efficiency; however, in practice, the trade-offs of higher variance and potential instability or divergence make implementing off-policy methods challenging and problematic [25, Sec. 11.10]. On-policy methods, on the other hand, are often more practical because of their ease of implementation and greater empirical stability.

Online and offline algorithms The distinction between *online* and *offline* methods is related but orthogonal to the on-policy/off-policy methods division. Whereas the on-policy/off-policy distinction concerns the source of the interaction data, online and offline methods are differentiated by *how* the data are collected. In online methods, the agent learns while it is actively interacting with the environment—the data collection and the learning process are interleaved in time. In offline methods (or batch mode algorithms [25, p. 452]), by contrast, the interaction dataset is fixed and has been collected prior to the learning process.

In an offline setup, therefore, the agent must learn a policy without being able to interact with the environment and obtain new experiences. This constraint introduces a set of unique challenges, including the inability to perform additional online exploration and the lack of observable counterfactual outcomes for actions that were not taken [17, Sec. 2.4]. In practice, the division between online and offline methods is often less a matter of design choice and more a restriction imposed by the problem setting. It may be impractical, dangerous, or unethical to apply online methods to real-world problems in domains where collecting and experimenting with new data is expensive or risky (such as in healthcare or industrial control systems); in such scenarios, offline methods are often the most viable way to apply RL.

2.2 Agent-based modeling

Agent-based modeling is a modeling method that uses computer simulations of many distinct entities, each following specified behavioral rules, to study how their interactions can produce complex overall patterns that are not directly imposed at the aggregate level. Beyond being a method, agent-based modeling can also be understood as a mindset for describing and understanding systems in terms of their constituent parts [2]. This bottom-up perspective allows agent-based modeling to investigate *emergent* phenomena—system-level behaviors that are counterintuitive and difficult

to predict from descriptions of individual units alone [22, Ch. 8].

Agents, environment and interactions Agent-based models (ABMs) are composed of individual decision-making entities, termed *agents*. In this context, an agent is a modeled entity and does not necessarily correspond to a learning agent in the RL sense. In ABMs, agents are often *heterogeneous*, differing from each other in attributes (such as size or location), which are often represented as internal state variables. Moreover, agents act *autonomously*, following rules or pursuing objectives that may differ across agents and are not necessarily aligned with one another.

Representing heterogeneity and autonomy explicitly is one of the key features distinguishing agent-based modeling from aggregate equation-based approaches. Aggregate models typically require *aggregation assumptions*: that differences across individuals "average out" into a tractable relationship at the system level. However, in systems where outcomes are the consequences of how the agent attributes are distributed (not just their mean), interactions can amplify small individual-level (or micro-level) differences into system-level (or macro-level) patterns that are difficult to fit into smooth equations. By treating system-level behavior as the consequence of decentralized decision-making, agent-based modeling can relax some aggregation assumptions and represent heterogeneity explicitly, at the cost of added modeling and computational complexity.

Typically, interactions between agents are *local*—agents interact with only a subset of all agents and can access only a limited subset of the information available in the model. This locality is shaped by the *environment* in which agents are situated, such as a geographical space or a network, and can make the modeling of agent interactions more explicit and natural, since in most real-world scenarios decision-makers operate under imperfect information and consider only a limited set of options.

Time, scheduling and stochasticity ABMs evolve over the simulation period through repeated updates of agent states and interactions, which can occur simultaneously, sequentially or according to a randomized activation schedule. Different schedules can lead to different model outcomes because the actions of one agent can change the environment encountered by other agents activated later. Relatedly, ABMs also commonly include stochasticity in their behavioral rules, interactions or initial conditions. As a result, repeated simulations even with the same model parameters can produce different trajectories, often making multiple runs necessary for assessing the consistency and variability of model outcomes [22, Chs. 14–15].

Complexity and emergence ABMs are often used to study *complex systems*. Whereas agent-based modeling is a modeling method, as well as a way of reasoning about systems from the bottom up, the study of complex systems (or *complexity science*) is a broader, transdisciplinary field that investigates phenomena across domains, including biology, physics and economics. Although there is no universal agreement on the definition of complexity or the methods of measuring complexity, a practical description of a complex system is a system that exhibits sophisticated macroscopic patterns, information processing and adaptation arising from networks of microscopic components following relatively simple rules of operation, without a central controller [18, Ch. 1]. Because complex systems are often analytically or mathematically intractable, computer simulations are useful tools for investigating possible underlying mechanisms and reproducing patterns observed in real-world systems [18, Ch. 14, 2].

One of the key properties of complex systems studied through agent-based modeling is *emergence*—the appearance of complex and often unpredictable macro-level patterns arising from relatively simple micro-level mechanisms. Similar to complexity, there is no straightforward way to measure the degree of emergence of certain phenomena; however, qualitative evaluation benchmarks can be applied when analyzing emergent macro-level patterns, such as how distinct they are from micro-level properties and how predictable they are given the underlying micro-level rules of operation [22, Ch. 8]. Although the concept of emergence may appear subjective, one goal of agent-based modeling is to explain emergent phenomena by simulating micro-level interactions. If an agent-based model can reproduce a real-world emergent phenomenon, it can provide a candidate explanation of the process that gives rise to it. Nevertheless, reproducing an observed pattern alone does not establish that the model contains the correct underlying mechanism.

It is worth noting, however, that the usefulness of a model does not increase simply with the degree of emergence it produces. A model whose emergent behavior arises from overly complex micro-level mechanisms can become intractable and difficult to learn from; a central goal of modeling is to understand cause-and-effect relationships, and emergence alone is not a sufficient explanation of real-world phenomena.

Decision-making behavior As agent-based modeling aims to understand macro-level patterns as a consequence of micro-level decision-making, the modeling of *agent behavior*—how agents act during a simulation—is a fundamental step in developing an agent-based model [22, Ch. 11]. Agent behavior is described using behavior

submodels, which specify the rules that agents follow based on their internal state variables and externally observable inputs. These submodels can range from simple if–then rules to more detailed *adaptive* mechanisms—that is, mechanisms in which the decision-making procedure itself changes over time in response to changes in the environment or the agent's internal state. Because different behavioral specifications can produce different systemic consequences, designing and justifying agent behavior is one of the central challenges in agent-based modeling.

2.3 RL in agent-based computational economics (ACE)

Agent-based computational economics is an area of computational economics that studies economic processes modeled as evolving systems of autonomous interacting agents, such as households, firms and institutions [26]. In essence, ACE applies agent-based modeling to the study of economic systems. The economic model used in this thesis is a stylized and parsimonious macroeconomic example of this approach.

Methodologically, machine learning and agent-based modeling can be integrated in several complementary ways, including the learning of agent behavior and sequential decision-making within simulated environments [28]. Within economics and finance, RL has been applied to a wide range of decision problems [5]. Several recent studies have examined the intersection of these research areas. The AI Economist developed by Zheng et al. [29] is a prominent example in which both individual economic agents and a social planner learn within a simulated economy, with the planner designing taxation policies. Dwarakanath et al. [7] more recently introduced ABIDES-Economist, a configurable agent-based economic simulation environment that supports both rule-based agents and agents whose policies are learned through RL.

Most directly relevant to this thesis, Brusatin et al. [3] studied the effects of introducing profit-oriented RL firms into a macroeconomic ABM. Their results show that the learned firms develop distinct profit-maximizing strategies, with the preferred strategy depending on market competition and the share of RL firms in the economy. Increasing the number of RL firms improves total economic output in their simulations, although the effect on economic instability depends on the strategy learned by the firms. Collectively, these studies demonstrate that RL can be incorporated into agent-based economic simulations at different levels, from individual economic behavior to policy design, and that the resulting system-level effects depend on both the learned behavior and the structure of the simulated economy.

2.4 Proximal Policy Optimization (PPO)

Proximal Policy Optimization is an on-policy actor-critic method for training parameterized stochastic policies [24]. PPO was developed to improve the stability of policy-gradient learning by limiting the incentive for excessively large policy updates. Unlike methods that impose an explicit constraint on the size of the policy update, PPO uses a simpler surrogate objective that discourages updates from moving the new policy too far from the policy used to collect the training data. Its relative implementation simplicity, empirical stability and applicability to both discrete and continuous action spaces have contributed to its widespread use in reinforcement learning.

2.4.1 Policy-gradient and actor-critic methods

Briefly introduced in Section 2.1.2, policy-based methods aim to learn a parameterized policy directly instead of deriving the policy from an estimated value function. Let $\theta \in \mathbb{R}^{d'}$ denote the parameters of the policy. The objective of policy-based methods is then to find values of θ that maximize a scalar performance measure $J(\theta)$, most commonly the expected discounted return, so that

$$J(\theta) \doteq \mathbb{E}_{\pi_\theta}[G],$$

where π_θ denotes the policy with parameters θ, G is the return defined in Equation (2), and the expectation is taken over trajectories generated by π_θ. *Policy-gradient* methods are a subset of policy-based methods that improve the policy by performing a gradient ascent update step on θ, such that

$$\theta \leftarrow \theta + \alpha \widehat{\nabla_\theta J(\theta)},$$

where $\widehat{\nabla_\theta J(\theta)}$ approximates the gradient of the performance measure $J(\theta)$ and α is the update step size.

For notational convenience, let π denote π_θ and let all gradients be taken with respect to θ. Using the policy-gradient theorem and the log-derivative rule, this gradient can be expressed in a form that can be estimated from sampled interactions with the environment [25, Secs. 13.2–13.4]:

$$\nabla J(\theta) \propto \mathbb{E}_\pi [\nabla \log \pi(A_t \mid S_t) q_\pi(S_t, A_t)],$$

where the term $\nabla \log \pi(A_t \mid S_t)$ describes how the log-likelihood of the selected action

changes with the parameters of the policy, while $q_\pi(S_t, A_t)$ weights this change based on the expected return of the action. Intuitively, the update changes the likelihood of each sampled action in proportion to its estimated value. Because the gradient-ascent update depends on the ascent direction and its effective step size, the positive proportionality factor can be absorbed into α.

In practice, however, using sampled returns or action-value estimates directly can produce high-variance gradient estimates, leading to inefficient learning [25, Sec. 13.4]. To address this, a state-dependent baseline can be subtracted from the sampled return or action-value estimate. This difference is represented by the *advantage function*:

$$\mathrm{Adv}_\pi(s, a) \doteq q_\pi(s, a) - v_\pi(s).$$

A positive advantage indicates that an action is better than the policy's expected outcome in the current state, whereas a negative advantage indicates that it is worse. Thus, using $v_\pi(s)$ as a baseline for the sampled returns can reduce the variance of the gradient estimate without changing its expected policy gradient. A sample-based gradient estimator can therefore be written as

$$\hat{g} = \widehat{\mathbb{E}}_t \left[\nabla \log \pi(A_t \mid S_t) \widehat{\mathrm{Adv}}_t \right],$$

where $\hat{g}$ estimates $\nabla J(\theta)$, $\widehat{\mathrm{Adv}}_t$ estimates $\mathrm{Adv}_\pi(S_t, A_t)$, and $\widehat{\mathbb{E}}_t$ denotes an empirical average over sampled time steps.

Actor-critic methods differ from policy-gradient-only methods in the role assigned to the value function [25, Sec. 13.5]. In a policy-gradient-only method, the policy update is based directly on sampled returns. A state-value estimate $v_\pi(s)$ may be subtracted as a baseline to reduce the variance of the gradient estimator, but the sampled return remains the primary learning signal. In an actor-critic method, by contrast, the learned value function is used to construct a bootstrapped advantage estimate and therefore directly contributes to the learning signal used to update the policy. The value-function estimator is therefore called the *critic*, while the parameterized policy π_θ, which determines the action distribution, is called the *actor*. PPO follows this actor-critic structure, and its advantage estimation procedure is described in Section 2.4.3.

Treating the advantage estimates as fixed during the policy update, the sample-based gradient estimator can be obtained by differentiating the empirical policy-gradient objective

$$L^{PG}(\theta) = \widehat{\mathbb{E}}_t \left[\log \pi_\theta(A_t \mid S_t) \widehat{\mathrm{Adv}}_t \right].$$

Differentiating $L^{PG}(\theta)$ with respect to θ yields the estimator $\hat{g}$. While it is possible to use this objective to directly optimize the policy, repeatedly maximizing it using the same batch of observations can produce excessively large policy changes and degrade performance [24]. PPO addresses this problem by replacing the basic policy-gradient objective with a surrogate objective that discourages large changes relative to the policy that generated the observations.

2.4.2 The clipped surrogate objective

Because the candidate policy is optimized using trajectories and advantage estimates generated by the previous policy, the contribution of each sampled action is reweighted according to its likelihood under the candidate policy relative to the data-generating policy. Let θ_{old} denote the policy parameters used to collect the samples. This relative weighting is expressed by the probability ratio

$$r_t(\theta) \doteq \frac{\pi_\theta(A_t \mid S_t)}{\pi_{\theta_{\text{old}}}(A_t \mid S_t)}.$$

A ratio of $r_t(\theta) = 1$ means that the sampled action has the same likelihood under both policies. A ratio above one means that the updated policy assigns a higher likelihood to the action, whereas a ratio below one means that it assigns a lower likelihood. As the candidate policy moves away from the data-generating policy, this ratio changes the weighting of the sampled terms and consequently the direction and magnitude of the surrogate gradient.

Incorporating this ratio into the objective yields the *conservative policy iteration surrogate objective*

$$L^{CPI}(\theta) \doteq \widehat{\mathbb{E}}_t \left[r_t(\theta) \widehat{\mathrm{Adv}}_t \right].$$

Notably, the gradient of this surrogate objective coincides with that of the empirical policy-gradient objective at the beginning of the update, when $\theta = \theta_{\text{old}}$. Treating the advantage estimates as fixed, and since

$$\nabla r_t(\theta) = r_t(\theta) \nabla \log \pi_\theta(A_t \mid S_t),$$

and $r_t(\theta_{\text{old}}) = 1$, it follows that

$$\nabla L^{CPI}(\theta)|_{\theta=\theta_{\text{old}}} = \widehat{\mathbb{E}}_t \left[\nabla \log \pi_\theta(A_t \mid S_t) \widehat{\text{Adv}}_t \right] = \nabla L^{PG}(\theta)|_{\theta=\theta_{\text{old}}}.$$

Thus, $L^{CPI}(\theta)$ and the empirical policy-gradient objective have the same initial gradient. As the candidate policy moves away from the data-generating policy, the probability ratio reweights the sampled terms, and their gradients generally differ.

PPO modifies the surrogate objective by clipping the probability ratio [24]:

$$L^{CLIP}(\theta) \doteq \widehat{\mathbb{E}}_t \left[\min \left(r_t(\theta) \widehat{\text{Adv}}_t, \text{clip}(r_t(\theta), 1-\epsilon, 1+\epsilon) \widehat{\text{Adv}}_t \right) \right],$$

where $\epsilon > 0$ is a clipping hyperparameter and $\text{clip}(x, a, b) = \min\{\max\{x, a\}, b\}$. When the estimated advantage is positive, the clipped objective removes the incentive to increase the probability ratio beyond $1 + \epsilon$. When the estimated advantage is negative, it removes the incentive to reduce the ratio below $1 - \epsilon$. The minimum of the unclipped and clipped terms produces a pessimistic estimate of the potential policy improvement, preventing clipping from artificially increasing the objective. Importantly, clipping does not impose a strict constraint requiring the probability ratio to remain within the interval $[1 - \epsilon, 1 + \epsilon]$. Instead, it only limits the improvement in the objective obtainable from moving the ratio beyond this interval in the advantageous direction.

2.4.3 Advantage estimation and the PPO objective

The clipped surrogate objective defined in the previous subsection requires an estimate of the advantage for each sampled state-action pair. In actor-critic methods, these estimates are typically constructed using the state-value function approximated by the critic. Let $\hat{v}(s)$ denote the estimate of the state-value function. A one-step measure of the error in the critic's prediction is given by the *temporal-difference residual*

$$\delta_t \doteq R_{t+1} + \gamma \hat{v}(S_{t+1}) - \hat{v}(S_t),$$

where γ is the discount rate introduced in Section 2.1.1. The temporal-difference residual compares the value predicted for state S_t with a one-step target formed by the sum of the observed reward and the discounted estimated value of the subsequent state. A positive residual indicates that the outcome was better than the critic predicted, whereas a negative residual indicates that it was worse. If S_{t+1} is a terminal state, $\hat{v}(S_{t+1})$ is defined as zero.

Although the temporal-difference residual may itself be used as a one-step estimate

of the advantage function, this estimate can be biased when the critic does not accurately approximate the true state-value function. Longer-horizon estimates can reduce this dependence on the critic by incorporating additional observed rewards. A k-step advantage estimate can be given by

$$\widehat{\mathrm{Adv}}_t^{(k)} \doteq \sum_{l=0}^{k-1} \gamma^l \delta_{t+l}.$$

As k increases, the estimate uses more observed rewards and relies less heavily on the critic. This can reduce the bias introduced by an inaccurate value function, but it also increases variance because the estimate accumulates randomness over a longer trajectory. PPO commonly uses Generalized Advantage Estimation (GAE), which addresses this trade-off by taking an exponentially weighted sum of the residual estimates [23]. The GAE estimate is defined as

$$\widehat{\mathrm{Adv}}_t^{\mathrm{GAE}} \doteq \sum_{l=0}^{\infty} (\gamma\lambda_{\mathrm{GAE}})^l \delta_{t+l},$$

where $\lambda_{\mathrm{GAE}} \in [0, 1]$ controls how strongly residuals from more distant steps contribute to the estimate. The one-step temporal-difference residual is a special case of GAE when $\lambda_{\mathrm{GAE}} = 0$. As λ_{GAE} approaches one, greater weight is assigned to longer-horizon returns, reducing the effect of value-function approximation error at the cost of increased variance. GAE therefore provides a mechanism for controlling the bias-variance trade-off in the advantage estimates.

The expression above gives the general definition of GAE over the remaining trajectory. In practical PPO training, data are commonly collected in fixed-length rollout segments that may end before the episode terminates. For a sampled sequence ending at time K, the corresponding truncated estimator is

$$\widehat{\mathrm{Adv}}_{t,K}^{\mathrm{GAE}} \doteq \sum_{l=0}^{K-t-1} (\gamma\lambda_{\mathrm{GAE}})^l \delta_{t+l},$$

where K denotes the final state index of the sampled sequence. When the sampled sequence ends because a terminal state is reached, the value beyond the terminal state is set to zero. When data collection ends before the episode terminates, the critic's estimate of the final observed state is used to bootstrap the remaining value. This allows GAE to be calculated from fixed-length rollout segments without requiring each segment to contain a complete episode.

In PPO and actor-critic methods, the critic is often represented by a parameterized

function, similar to how the actor is represented. In neural-network implementations, the actor and critic may use separate networks or share an initial feature-extraction network before branching into separate outputs. The implementation used in this thesis follows the latter, as detailed in Section 3.3.2. When parameters are shared, gradients from both the policy and value objectives contribute to updating the shared representation.

A target for training the critic can be formed naturally from the estimated advantage used in the policy objective as

$$V_t^{\text{targ}} \doteq \widehat{\text{Adv}}_t + \hat{v}_{\text{old}}(S_t),$$

where $\hat{v}_{\text{old}}(S_t)$ denotes the value estimate used when the advantage estimates were calculated. Treating V_t^{targ} as fixed during the update, the critic can be trained by minimizing the squared-error value-function loss given by

$$L^V \doteq \widehat{\mathbb{E}}_t \left[\left(\hat{v}(S_t) - V_t^{\text{targ}} \right)^2 \right].$$

Reducing this loss improves the critic's ability to predict future returns and consequently the advantage estimates supplied to the actor.

The clipped surrogate policy objective and the value-function loss can be combined into an overall PPO objective. This overall objective may also include an entropy bonus to encourage exploration during training, so that

$$L^{PPO} \doteq L^{CLIP} - c_V L^V + c_H \widehat{\mathbb{E}}_t \left[\mathcal{H}[\pi_\theta](S_t) \right],$$

where c_V and c_H are coefficients determining the contributions of the value-function loss and entropy bonus respectively, and $\mathcal{H}$ denotes the entropy of the policy's action distribution in state S_t. This expression is written as an objective to be maximized; when PPO is implemented as the minimization of a loss function, the signs of the terms are reversed.

3 Methods

3.1 The Lengnick baseline model (LBM)

The Lengnick baseline model is a minimal ACE model designed to explore the macrodynamics of a closed, zero-growth economy consisting of only two kinds of agents with simple adaptive rules. It follows the bottom-up approach strictly: the interactions between agents are decentralized and the information the agents observe is local. Despite its simplicity, this model has demonstrated its capability of reproducing several empirical stylized facts, including endogenous business cycles, Phillips and Beveridge curves, and realistic unemployment rates. This section is based primarily on Lengnick [16].

3.1.1 Agents and state variables

The model has two types of agents: households and firms. Households are indexed by $h = 1, 2, \ldots, H$ and firms by $f = 1, 2, \ldots, F$. The state variables and corresponding properties of each agent type are described in Table 1.

Table 1: State variables and corresponding properties of household and firm agents in the model.

Agent	State variable	Property name	Description
Household	m_h	Liquidity	Number of monetary units held
	ω_h	Reservation wage	Minimum acceptable monthly wage
Firm	m_f	Liquidity	Number of monetary units held
	i_f	Inventory	Quantity of produced consumption goods
	p_f	Goods price	Price per unit of consumption good
	w_f	Wage	Monthly wage paid per employee

The liquidity properties m_h and m_f determine the financial positions of households and firms respectively, and are updated whenever an agent conducts a transaction—households when spending and receiving income, and firms when making sales, paying wages, and distributing profits. The reservation wage ω_h determines the

minimum monthly wage a household is willing to accept and is readjusted based on the household's employment status and job search progress. Inventory i_f is updated as the firm produces, while goods price p_f and wage rate w_f are set by firms.

3.1.2 Markets, interaction structure and information

The model economy has two markets: the consumption goods market and the labor market. In both markets, transactions occur between a firm and a household. These interactions are governed by a network structure (the environment) with two types of connections: trading (Type A) and employment (Type B) connections.

The goods market A household can purchase goods from a firm to which it has a trading relation, provided the household has sufficient liquidity and the firm has sufficient inventory. The network structure limits the number of trading connections (or Type A connections) a household can have. This reflects the empirical observation that consumers exhibit some degree of loyalty to particular brands. Specifically, in the LBM and the extended model of this thesis, a household agent has seven Type A connections and can observe price information only from those seven firms. The only exception to this restriction occurs when households search for better suppliers: once per search operation, they observe the price of a firm sampled with probability proportional to its size and compare it with the prices of their current suppliers.

Firms are not limited in the number of Type A connections they can have. However, firms also operate under local information: they do not observe competitors' prices.

The labor market To produce consumption goods, firms must hire households as workers. In exchange for their labor, employed households receive a monthly wage paid by their employers. A household may have only one employment connection (or Type B connection) with a single firm, while there is no limit on how many employees a firm can have. A Type B connection is established if a firm has an open position and offers a wage that meets the requirements of a household looking for employment. During their employment search, households can observe wage offers from only a limited number of potential employers. The number of firms they consider depends on their search intensity, which varies with employment status and with the relationship between their current wage and reservation wage.

As in the goods market, firms have only local information in the labor market and do not observe wage information from their competitors. Details on what firms can observe are specified in Section 3.1.5.

3.1.3 Timing, process overview and scheduling

The processes in the LBM are divided into two categories: monthly processes and daily processes. This division reflects the fact that different activities occur at different frequencies in real-world settings. In both the LBM and the extended model, a month consists of 21 working days. Daily processes include consumption and production, while wage/price decisions and labor-market search are handled monthly. The process schedule is summarized in Table 2.

Table 2: The process schedule of the model. Processes are executed in the order shown (top to bottom). Daily processes are repeated on each of the 21 working days per month.

Phase (category)	Entities	Agent execution order	Process
Beginning of month (monthly)	Firms	—	Set wage
		—	Manage workforce (hiring/firing decision)
		—	Set goods price
	Households	—	Search for better suppliers
		Random	Find employment
		—	Budget consumption
During month (daily)	Households	Random	Purchase goods
	Firms	—	Produce goods
End of month (monthly)	Firms	—	Pay wages
		—	Build liquidity buffer
		—	Distribute profits
	Households	—	Adjust reservation wage

For most processes, the order of execution within an agent type does not matter due to the independent nature of the process. However, ordering can matter for certain interactions— job matching (vacancies are limited) and goods purchasing (inventories are limited)—because earlier-activated households may obtain opportunities that are no longer available to later-activated households.

3.1.4 Household behavior

Search for better suppliers Each household h maintains a set of Type A connections $A_h \subseteq \{1, \ldots, F\}$ with a fixed size $n_A = 7$. At the beginning of each month, each household may search for better suppliers and update its supplier set A_h if improvements are found. The search for suppliers consists of a *price-based search* and a *demand-based search*.

A price-based search is conducted with a probability of $\psi_{\text{price}} \in (0, 1)$. The household first randomly selects a current supplier $f^{\text{old}}_{\text{price}} \in A_h$, with

$$\Pr(f^{\text{old}}_{\text{price}} = i) = \frac{1}{n_A}, \quad \forall i \in A_h.$$

It then samples a candidate replacement firm $f^{\text{new}}_{\text{price}} \notin A_h$, weighted by firm size (measured by the number of employees). Let $B_f \subseteq \{1, \ldots, H\}$ denote the set of households employed by firm f, so that the sampling probability for a replacement firm is

$$\Pr(f^{\text{new}}_{\text{price}} = j) = \frac{|B_j| + 1}{\sum_{\ell \in \{1,\ldots,F\} \setminus A_h} (|B_\ell| + 1)}, \quad \forall j \in \{1, \ldots, F\} \setminus A_h.$$

Importantly, firm size $|B_f|$ can be zero, which is problematic when used as a sampling weight because it implies that firms with no employees are never sampled as replacement firms under the size-weighted rule. This invisibility leads to a self-reinforcing downward spiral where once a firm has no employees, it cannot acquire new customers even if it posts a low price despite holding inventory. To avoid this issue and keep the model economy stable, the implementation uses the adjusted weight $|B_f| + 1$ in the sampling distribution.

The household replaces the Type A connection if the candidate firm's price is sufficiently lower than the price of the current supplier. Using a price threshold $\xi \in (0, 1)$, the updated supplier set is

$$A_h \leftarrow (A_h \setminus \{f^{\text{old}}_{\text{price}}\}) \cup \{f^{\text{new}}_{\text{price}}\}, \quad \text{if } p_{f^{\text{new}}_{\text{price}}} \leq (1 - \xi) p_{f^{\text{old}}_{\text{price}}}.$$

Otherwise, the supplier set remains unchanged.

After the price-based search, a demand-based search is conducted with a probability of $\psi_{\text{quant}} \in (0, 1)$. In addition, the demand-based search occurs only if the household's consumption need has not been met by at least one firm during the previous month—the firm did not have enough inventory despite the household having sufficient liquidity. Among the demand-constrained suppliers, the household randomly selects one firm to

replace, weighted by the degree of constraints (measured by the sum total of unsatisfied demand). Let $u_{h,f} \geq 0$ denote the total quantity of unsatisfied demand that household h experienced at firm f during the previous month; the set U_h of demand-constrained suppliers for household h is

$$U_h = \{f \in A_h : u_{h,f} > 0\}.$$

If $U_h \neq \emptyset$, the probability that firm $f^{\text{old}}_{\text{quant}}$ would be selected for replacement is

$$\Pr(f^{\text{old}}_{\text{quant}} = i) = \frac{u_{h,i}}{\sum_{\ell \in U_h} u_{h,\ell}}, \quad \forall i \in U_h.$$

The replacement supplier $f^{\text{new}}_{\text{quant}} \notin A_h$ is then drawn uniformly at random from firms that do not have Type A connections to the household,

$$\Pr(f^{\text{new}}_{\text{quant}} = j) = \frac{1}{F - n_A}, \quad \forall j \in \{1, \ldots, F\} \setminus A_h.$$

The household then updates its supplier set through

$$A_h \leftarrow (A_h \setminus \{f^{\text{old}}_{\text{quant}}\}) \cup \{f^{\text{new}}_{\text{quant}}\}.$$

After the demand-based search is complete, the set of demand-constrained firms U_h is reset for the coming month, i.e. $U_h \leftarrow \emptyset$.

Together, the price-based search and the demand-based search represent the consumers' preference for cheaper goods and reliable suppliers, with the implication that the demand a firm encounters is negatively correlated with its price and its failure to satisfy past demand.

Find employment At the beginning of each month and in a random order, households participate in the labor market, trying to find employment if unemployed or looking for better wage offers otherwise. The random activation order matters because a firm can only open *at most* one new position each month (detailed in Section 3.1.5), and a household filling an open position will deny another household's chance of establishing an employment connection with the same firm that month. While a firm can have more than one employee, a household can only be employed by one employer so that

$$B_f \cap B_{f'} = \emptyset, \quad \forall f \neq f'.$$

Depending on the current employment status, current wage and current reservation wage, households differ in their job search intensity, quantified by the number of search attempts n_h^{att} and the probability π_h of the job search process being triggered. Let w_h^{cur} denote the current wage household h receives each month,

$$w_h^{\text{cur}} = \begin{cases} w_f, & \text{if } \exists f \in \{1, \dots, F\} \text{ such that } h \in B_f, \\ 0, & \text{otherwise,} \end{cases}$$

where $w_h^{\text{cur}} = 0$ indicates unemployment. Households can then be divided into three cases: employed and satisfied with their current wage, employed and dissatisfied with their current wage, and unemployed. These cases determine the household's search intensity.

The number of search attempts is given by

$$n_h^{\text{att}} = \begin{cases} 1, & \text{if } w_h^{\text{cur}} > 0, \\ \beta, & \text{otherwise,} \end{cases}$$

where β denotes the number of job search attempts made by an unemployed household.

The probability of the household engaging in a job search is

$$\pi_h = \begin{cases} \pi_{\text{emp}}, & \text{if } w_h^{\text{cur}} \geq \omega_h, \\ 1, & \text{otherwise,} \end{cases}$$

where $\pi_{\text{emp}} \in (0, 1)$ is the job-search probability of an employed household whose current wage is greater than or equal to its reservation wage. Therefore, unemployed households and employed households dissatisfied with their current wages always search, whereas satisfied employed households search only with probability π_{emp}.

If the job search process is triggered, household h performs n_h^{att} search attempts, in each of which the household samples one candidate firm uniformly at random. For unemployed households, the candidate set is all of the firms, whereas for employed households, the current employer is excluded from the set. The candidate employer set for household h can then be denoted as

$$\mathcal{F}_{h,\,\text{job}} = \{f \in \{1, \dots, F\} : h \notin B_f\}.$$

If $n_h^{\text{att}} > 1$, firms are sampled without replacement over the household's search attempts to avoid repeated visits to the same candidate employer. Let $f_{h,\,\text{job}}^{(k)}$ denote the firm visited in attempt k, the remaining candidate set before attempt k is

$$\mathcal{F}_{h,\,\text{job}}^{(k)} = \begin{cases} \mathcal{F}_{h,\,\text{job}}, & \text{if } k = 1, \\ \mathcal{F}_{h,\,\text{job}} \setminus \{f_{h,\,\text{job}}^{(1)}, \ldots, f_{h,\,\text{job}}^{(k-1)}\}, & \text{if } k \geq 2, \end{cases}$$

and the sampling probability in attempt k is

$$\Pr(f_{h,\,\text{job}}^{(k)} = j) = \frac{1}{|\mathcal{F}_{h,\,\text{job}}^{(k)}|}, \quad \forall j \in \mathcal{F}_{h,\,\text{job}}^{(k)}.$$

If the visited firm $f_{h,\,\text{job}}^{(k)}$ has an open position and offers an acceptable wage, a match is formed. An unemployed household will accept an offer if the offered wage is at least as high as its reservation wage, whereas an employed household will only make the switch if it can improve its current wage. Notably, in Lengnick [16], the job search procedure is described only in terms of comparisons to the household's current wage, and the role of the reservation wage in the household's decision to accept a job offer is not made explicit. Given the description of the reservation wage in the original paper and its update rules presented later in this section, the implementation in this thesis uses reservation wage as the acceptance threshold for unemployed households.

Let $v_f \in \{0, 1\}$ indicate whether firm f has a vacancy, where $v_f = 1$ denotes an open vacancy and $v_f = 0$ otherwise. The update rules for v_f are detailed in Section 3.1.5. A Type B connection is established immediately if the match is suitable, as formalized by the following update rule:

$$B_{f_{h,\,\text{job}}^{(k)}} \leftarrow \begin{cases} B_{f_{h,\,\text{job}}^{(k)}} \cup \{h\}, & \text{if } v_{f_{h,\,\text{job}}^{(k)}} = 1 \text{ and } w_h^{\text{cur}} = 0 \text{ and } w_{f_{h,\,\text{job}}^{(k)}} \geq \omega_h, \\ B_{f_{h,\,\text{job}}^{(k)}} \cup \{h\}, & \text{if } v_{f_{h,\,\text{job}}^{(k)}} = 1 \text{ and } w_{f_{h,\,\text{job}}^{(k)}} > w_h^{\text{cur}} > 0, \\ B_{f_{h,\,\text{job}}^{(k)}}, & \text{otherwise.} \end{cases}$$

If household h is employed and making a job switch, the Type B connection between it and its current firm f_h^{cur} with $h \in B_{f_h^{\text{cur}}}$ is dissolved first:

$$B_{f_h^{\text{cur}}} \leftarrow B_{f_h^{\text{cur}}} \setminus \{h\}.$$

After a successful match, the vacancy of the hiring firm is closed for the remainder of the month,

$$v_{f_{h,\,\text{job}}^{(k)}} \leftarrow 0,$$

and the household stops searching.

If none of the sampled firms has an open position or offers an acceptable wage during the n_h^{att} search attempts, the household's employment status remains unchanged for that month.

Budget consumption After the job search process has been completed, each household determines its planned consumption expenditure for the coming month, with the remainder of its liquidity retained as savings. In the LBM, this decision depends on the household's real wealth, measured as current liquidity relative to the average goods price of the firms to which the household has Type A connections.

Let P_h^I denote the average goods price of household h's suppliers

$$P_h^I = \frac{\sum_{f \in A_h} p_f}{n_A}.$$

The planned consumption expenditure c_h^r is then given by

$$c_h^r = \min \left\{ \left(\frac{m_h}{P_h^I} \right)^{\alpha_{\text{elas}}}, \frac{m_h}{P_h^I} \right\},$$

where $\alpha_{\text{elas}} \in (0, 1)$ governs the elasticity of planned consumption with respect to real wealth. Since $0 < \alpha_{\text{elas}} < 1$, planned consumption is increasing in real wealth, but at a decreasing rate. The minimum operator ensures that the household's planned expenditure does not exceed the household's currently available real wealth, which happens when $m_h / P_h^I < 1$.

Importantly, this process determines the household's planned consumption for the month, but not the exact timing or allocation of purchases across firms. The actual realization of consumption takes place during the daily goods purchasing process, where the household attempts to satisfy its planned demand by making purchases from its suppliers.

Purchase goods After the monthly processes at the beginning of the month have been completed, households proceed to the daily processes that take place on each working day. On each day, households are activated in a random order and attempt to realize the consumption they planned earlier. Since planned consumption c_h^r is determined on a monthly basis, but transactions take place daily, c_h^r is assumed to be distributed evenly over the month, so that the daily consumption target of household h is

$$d_h = \frac{c_h^r}{21},$$

where a month consists of 21 working days. The random activation order matters because firm inventories are limited: a household that makes a purchase earlier reduces the inventory available to households that visit the same firm later.

To satisfy its consumption demand, household h visits firms from its supplier set A_h uniformly at random without replacement. If the visited firm does not have enough inventory to satisfy the household's remaining demand, the household may continue to another supplier that has not been visited previously. This process continues until the household's demand has been sufficiently met, all firms in A_h have been visited, or the household has exhausted its liquidity.

Let $f_{h,\,\mathrm{sup}}^{(k)}$ denote the firm visited in purchasing attempt k, and let $d_h^{(k)}$ denote the remaining consumption demand of household h before attempt k, with $d_h^{(1)} = d_h$. To account for the household's liquidity constraint, let $\tilde{d}_h^{(k)}$ denote the household's *effective* demand in attempt k, that is, the remaining demand truncated by the household's available liquidity, given by

$$\tilde{d}_h^{(k)} = \min\left\{ d_h^{(k)}, \frac{m_h}{p_{f_{h,\,\mathrm{sup}}^{(k)}}} \right\}.$$

The transaction quantity $q_h^{(k)}$ in attempt k is then determined by the effective demand $\tilde{d}_h^{(k)}$ and the available inventory at firm $f_{h,\,\mathrm{sup}}^{(k)}$:

$$q_h^{(k)} = \min\left\{ \tilde{d}_h^{(k)}, i_{f_{h,\,\mathrm{sup}}^{(k)}} \right\}.$$

If the firm does not have enough inventory to satisfy the household's effective demand, the unmet demand is recorded for the supplier evaluation process in the following month:

$$u_{h,f_{h,\,\mathrm{sup}}^{(k)}} \leftarrow u_{h,f_{h,\,\mathrm{sup}}^{(k)}} + (\tilde{d}_h^{(k)} - q_h^{(k)}), \quad \text{if } q_h^{(k)} < \tilde{d}_h^{(k)}.$$

The transaction updates liquidity and inventory according to

$$\begin{aligned} m_h &\leftarrow m_h - p_{f_{h,\,\mathrm{sup}}^{(k)}} q_h^{(k)}, \\ m_{f_{h,\,\mathrm{sup}}^{(k)}} &\leftarrow m_{f_{h,\,\mathrm{sup}}^{(k)}} + p_{f_{h,\,\mathrm{sup}}^{(k)}} q_h^{(k)}, \\ i_{f_{h,\,\mathrm{sup}}^{(k)}} &\leftarrow i_{f_{h,\,\mathrm{sup}}^{(k)}} - q_h^{(k)}. \end{aligned}$$

After the purchase attempt, the remaining daily demand is updated as

$$d_h^{(k+1)} = \tilde{d}_h^{(k)} - q_h^{(k)}.$$

The process is repeated for purchasing attempt $k+1$ until either 95% of the planned daily demand has been satisfied, i.e. $d_h^{(k+1)} \leq 0.05 d_h$, all suppliers have been visited, i.e. $k = n_A$, or the household exhausts its liquidity, i.e. $m_h = 0$.

Adjust reservation wage At the end of each month, households adjust their reservation wages based on their currently received wages, with the update rule given by

$$\omega_h \leftarrow \begin{cases} w_h^{\text{cur}}, & \text{if } w_h^{\text{cur}} > \omega_h, \\ \omega_h, & \text{if } 0 < w_h^{\text{cur}} \leq \omega_h, \\ 0.9\omega_h, & \text{if } w_h^{\text{cur}} = 0. \end{cases}$$

If a household's current labor income exceeds its reservation wage, the reservation wage is raised to the level of the received wage. If the household remains employed but receives a wage below its reservation wage, the reservation wage is not changed; instead, the household intensifies its search for a better-paid job in the next period. If the household has been unemployed during the month, its reservation wage is reduced by 10% for the following month.

Reservation wage is thus adapted over time and introduces persistent and endogenous frictions in the model's labor market. In particular, unemployed households gradually lower their reservation wages, which increases their willingness to accept lower-paid jobs over time. In this thesis, this supports the use of reservation wage as the acceptance threshold for unemployed households, although the original paper formulates job acceptance in terms of currently received wage rather than reservation wage explicitly.

3.1.5 Firm behavior

Set wage At the beginning of each month, each firm updates its wage offer based on its recent hiring success in the labor market. In the LBM, wage setting is purely adaptive and decentralized: firms do not observe competitors' wages, but react only to whether the wage they offered was sufficient to attract workers in the recent past. In particular, a firm raises its wage when it was unable to fill an open vacancy, and lowers

its wage when it has operated without an unfilled vacancy for a prolonged period.

Let $s_f \in \mathbb{N}$ denote the number of consecutive months during which firm f has been fully staffed, i.e. without an unfilled vacancy. The wage-setting rule is then given by

$$w_f \leftarrow \begin{cases} w_f(1+\mu_f), & \text{if } v_f = 1, \\ w_f(1-\mu_f), & \text{if } v_f = 0 \text{ and } s_f \geq \gamma_{\text{month}}, \\ w_f, & \text{otherwise,} \end{cases}$$

where μ_f is an idiosyncratic multiplicative adjustment rate drawn from a uniform distribution on $[0, \delta]$, i.e. $\mu_f \sim U(0, \delta)$, and γ_{month} is the threshold number of consecutive fully staffed months required before the firm lowers its wage.

The counter s_f is updated at the end of each month according to

$$s_f \leftarrow \begin{cases} 0, & \text{if } v_f = 1, \\ s_f + 1, & \text{if } v_f = 0. \end{cases}$$

The wage setting process in the LBM is thus a backward-looking heuristic: firms adjust wages only in response to their own recent vacancy outcomes, rather than by explicitly optimizing over any relevant performance metrics.

Manage workforce After setting wages, each firm determines whether to expand, reduce or maintain its workforce based on the relationship between its current inventory and the demand it faced during the previous month. The core idea is that current inventories reflect the firm's balance between production and sales: low inventories indicate that the firm may have produced too little and should increase hiring, whereas high inventories indicate overproduction and the need to downsize.

Let d_f denote the total demand encountered by firm f during the previous month, aggregated over household purchase attempts in the goods market, so that

$$d_f = \sum_{h=1}^{H} \sum_{k \in \mathcal{K}_{h,f}} \tilde{d}_h^{(k)},$$

where $\mathcal{K}_{h,f}$ is the set of purchase attempts in the previous month in which household h visited firm f. Thus, firm-level demand is defined here as the sum of effective demand directed at the firm before inventory rationing, rather than as realized sales.

Based on this demand, the firm forms upper and lower inventory thresholds,

$$\bar{i}_f = \overline{\phi}\, d_f,$$
$$\underline{i}_f = \underline{\phi}\, d_f,$$

where $0 < \underline{\phi} < \overline{\phi}$. The firm opens a new vacancy if the current inventory is below the lower threshold, i.e.

$$v_f \leftarrow \begin{cases} 1, & \text{if } i_f < \underline{i}_f, \\ 0, & \text{otherwise,} \end{cases}$$

and dismisses one worker chosen uniformly at random from its set of employees B_f if the current inventory exceeds the upper threshold. Provided that $|B_f| > 0$, the dismissed household is selected according to

$$\Pr(h_f^{\text{fire}} = j) = \frac{1}{|B_f|}, \quad \forall j \in B_f,$$

after which the selected worker is marked for dismissal at the beginning of the month, and the corresponding Type B connection is removed at the beginning of the following month, i.e.

$$B_f \leftarrow B_f \setminus \{h_f^{\text{fire}}\}.$$

If the current inventory lies between the two thresholds, the firm maintains its current workforce.

Importantly, hiring and firing processes are implemented asymmetrically. Hiring decisions take place immediately: vacancies opened at the beginning of the month can be filled during the same month. On the other hand, firing decisions are implemented with a one-month lag. The rationale for this asymmetry is the fact that workers are typically protected against immediate dismissal by employment protection laws and related institutional frictions.

The assumption that firms can adjust their workforce by at most one worker per month may appear restrictive at the individual-firm level. However, given the firm-to-household ratio of 1:10, it implies that aggregate labor turnover can still reach as much as 10% in a single month.

Set goods price After the workforce decision, firms proceed to evaluate their goods prices. Similar to workforce management, price setting is a mechanism for firms to adjust the balance between production and sales and is therefore conditional on the

firm's current inventory. However, whereas workforce management involves changes to production capacity, price setting adapts the firm's sales conditions to encountered demand.

The firm uses the same inventory thresholds defined in the workforce management process when considering whether to adjust its goods price. Low inventory, i.e. $i_f < \underline{i}_f$, indicates that demand has been high relative to supply and the firm should consider raising its price. Conversely, high inventory, i.e. $i_f > \bar{i}_f$, signals weak demand or excess supply, suggesting that the firm should lower its price. If the inventory lies between the lower and upper thresholds, the goods price is maintained.

It is assumed that the pricing process operates similarly to the Calvo pricing model [4], in that the goods price is sticky: a new price is implemented only with probability $\theta_{\text{calvo}} \in (0, 1)$, even if the firm has an unsatisfying level of inventory.

If the goods price is to be adjusted, upper and lower thresholds for the goods price are established to ensure the price stays within a reasonable range. Let mc_f denote the marginal cost of producing one unit of consumption good, i.e.

$$mc_f = \frac{w_f}{21\lambda},$$

where a month consists of 21 working days and λ is a positive technology parameter that dictates how many units an employee produces per day. The thresholds are derived from marginal cost according to

$$\begin{aligned} \overline{p}_f &= \overline{\varphi}\, mc_f, \\ \underline{p}_f &= \underline{\varphi}\, mc_f, \end{aligned}$$

where $1 < \underline{\varphi} < \overline{\varphi}$. Conditional on a Calvo-style price adjustment opportunity being realized, if the current goods price satisfies this threshold condition, it is updated according to

$$p_f \leftarrow \begin{cases} p_f(1 + \nu_f), & \text{if } i_f < \underline{i}_f \text{ and } p_f < \overline{p}_f, \\ p_f(1 - \nu_f), & \text{if } i_f > \bar{i}_f \text{ and } p_f > \underline{p}_f, \\ p_f, & \text{otherwise,} \end{cases}$$

where ν_f is an idiosyncratic multiplicative adjustment rate drawn from a uniform distribution on $[0, \vartheta]$, i.e. $\nu_f \sim U(0, \vartheta)$.

Similar to wage setting, the price setting process is local: firms do not observe competitors' prices or market-wide demand, but adjust their prices based solely on

their current inventory, previously encountered demand and production costs.

Produce goods During each working day of the month, firms produce consumption goods using their workforce. Production is deterministic and the output is a linear function of labor input. The firm's inventory is updated according to

$$i_f \leftarrow i_f + \lambda |B_f|,$$

where $\lambda > 0$ is the technology parameter defined above.

Pay wages At the end of each month, firms pay wages to their employees for their labor inputs. Under normal circumstances, each employee of firm f receives the promised wage w_f. However, in the case that the firm does not have enough liquidity to pay its workers, the firm cuts its wage to the maximum level it can afford to pay all of its employees equally, i.e.

$$w_f \leftarrow \begin{cases} w_f, & \text{if } m_f \geq w_f |B_f|, \\ \frac{m_f}{|B_f|}, & \text{if } m_f < w_f |B_f|. \end{cases}$$

The liquidity levels of the firm and its employees are then updated according to

$$\begin{aligned} m_h &\leftarrow m_h + w_f, \quad \forall h \in B_f, \\ m_f &\leftarrow m_f - w_f |B_f|. \end{aligned}$$

If the firm has no employees, no wage payment is made.

Build liquidity buffer After the payout of wages, the firm determines how much liquidity to retain as a buffer before distributing profits. The buffer serves as insurance against potential future losses that would prevent the firm from paying its workers the promised wage. As such, the target liquidity buffer is calculated relative to the firm's monthly labor costs, according to

$$m_f^{\text{buf}} \leftarrow \chi w_f |B_f|,$$

where $\chi \geq 0$. If the target buffer is larger than the firm's current liquidity, the target is readjusted to the maximum possible value, i.e. the firm's entire remaining liquidity:

$$m_f^{\text{buf}} \leftarrow \begin{cases} m_f^{\text{buf}}, & \text{if } m_f^{\text{buf}} \leq m_f, \\ m_f, & \text{if } m_f^{\text{buf}} > m_f. \end{cases}$$

The buffer is then subtracted from the firm's liquidity, i.e.

$$m_f \leftarrow m_f - m_f^{\text{buf}}.$$

Distribute profits After setting aside the buffer for bad times, the firm distributes any remaining liquidity as profits to all households. Each household receives a profit share that is proportional to its current liquidity. This is a simplifying proxy for stock ownership, allowing liquidity to circulate within the closed economy without introducing a more elaborate profit allocation structure.

Let M_H denote the total liquidity of all households before profit distribution, i.e.

$$M_H = \sum_{h=1}^{H} m_h.$$

Household liquidity is updated according to

$$m_h \leftarrow m_h + m_f \frac{m_h}{M_H}, \quad \forall h \in \{1, \ldots, H\},$$

where all terms on the right-hand side are evaluated before profit distribution. After profits have been distributed, the firm's liquidity is set to zero, i.e. $m_f \leftarrow 0$.

The month then ends, statistics are collected and the next month begins. At the start of the next month, for bookkeeping convenience, the liquidity buffer of each firm is transferred back into its liquidity:

$$\begin{aligned} m_f &\leftarrow m_f + m_f^{\text{buf}}, \\ m_f^{\text{buf}} &\leftarrow 0. \end{aligned}$$

3.1.6 Summary of variables and parameter values

Table 3 summarizes the auxiliary variables and mathematical notation introduced in the process descriptions, supplementing the core agent state variables listed in Table 1.

Table 4 lists the parameters of the LBM and their values used in this thesis. The values are based on the baseline calibration specified in [16].

Table 3: Auxiliary variables used in the LBM process descriptions.

Variable	Description
A_h	Set of suppliers of household h (Type A trading connections)
B_f	Set of households employed by firm f (Type B employment connections)
v_f	Vacancy indicator of firm f
s_f	Number of consecutive months during which firm f has operated without an unfilled vacancy
P_h^I	Average goods price among the suppliers of household h
c_h^r	Planned real monthly consumption of household h
d_h	Daily consumption target of household h
$\tilde{d}_h^{(k)}$	Effective demand of household h in purchase attempt k, after accounting for the household's liquidity constraint
$q_h^{(k)}$	Quantity purchased by household h in purchase attempt k
$u_{h,f}$	Unsatisfied demand experienced by household h at firm f
d_f	Total demand encountered by firm f during the previous month
$\underline{i}_f, \bar{i}_f$	Lower and upper inventory thresholds of firm f
mc_f	Marginal cost of producing one unit of consumption good for firm f
$\underline{p}_f, \overline{p}_f$	Lower and upper goods price thresholds of firm f
m_f^{buf}	Liquidity buffer of firm f
M_H	Total household liquidity before profit distribution

3.2 RL extension of the LBM

This thesis extends the LBM by introducing RL-controlled firms: firms whose wage, price and workforce decisions are produced by a learned policy. The other aspects of the LBM, including the market structure, process timing and household behavior, are preserved. The total number of firms in the model is also maintained, such that the extension replaces selected non-learning firms with RL firms instead of adding additional firms to the economy. Throughout the rest of the thesis, non-learning firms are referred to as baseline firms, equipped with the baseline firm behavior described in Section 3.1.5, to differentiate them from learned firms equipped with a learned policy.

3.2.1 Learning problem formulation

In the RL formulation, the learned firm is the RL agent and the LBM is the environment. The environment includes households and other firms, while the decision-making

Table 4: Baseline parameter values used in the model.

Parameter	Value	Description
H	1,000	Number of households
F	100	Number of firms
n_A	7	Number of Type A trading connections per household
α_{elas}	0.9	Elasticity of planned real consumption with respect to real wealth
ξ	0.01	Price threshold for replacing a supplier in the price-based search
ψ_{price}	0.25	Probability of conducting a price-based supplier search
ψ_{quant}	0.25	Probability of conducting a demand-based supplier search
β	5	Number of job search attempts made by an unemployed household
π_{emp}	0.1	Job search probability of an employed household satisfied with its current wage
γ_{month}	24	Number of consecutive fully staffed months required before a firm lowers its wage
δ	0.019	Upper bound of the idiosyncratic wage adjustment rate
$\underline{\phi}$	0.25	Lower inventory threshold multiplier
$\overline{\phi}$	1	Upper inventory threshold multiplier
$\underline{\varphi}$	1.025	Lower price threshold multiplier relative to marginal cost
$\overline{\varphi}$	1.15	Upper price threshold multiplier relative to marginal cost
ϑ	0.02	Upper bound of the idiosyncratic price adjustment rate
θ_{calvo}	0.75	Probability of implementing a newly determined goods price (Calvo stickiness)
λ	3	Number of consumption goods produced by one worker per day
χ	0.1	Fraction of labor costs the firm keeps as a buffer

rules of the learned firm are replaced by the learned policy. In the training setting used in this thesis, the firm population consists of one learned firm and 99 baseline firms. Simultaneously training multiple interacting learned firms would require a multi-agent RL formulation, which is outside the scope of this thesis. As such, the learned profit-maximizing behavior in this thesis is derived in an environment with heuristic competitors rather than other learning profit-maximizing firms.

A notion of profit maximization also needs to be specified. In this thesis, the objective of the learned firm is to maximize nominal profit, defined as goods-market revenue minus labor costs. Notably, this objective is different from other typical notions of profitability in an economic sense, such as optimizing margins or returns to investors through distributed profits. This design choice is made primarily for the simplicity of designing and implementing the reward function. Other notions of profitability, while somewhat related to nominal profit, represent different objectives and are discussed in more detail in Section 5.

Among the processes of firm behavior, the decisions most directly relevant to profit maximization are setting the goods price, setting the wage and managing the workforce. Producing goods and paying wages are not treated as learned actions: production is deterministic once workers have been hired, and wage payment is treated as an accounting obligation rather than a discretionary decision. Building the liquidity buffer and distributing profits are also not treated as learned actions. If these decisions were controlled by the learned policy, the firm could trivially retain liquidity by setting the buffer to the entire available liquidity or by distributing no profits to households.

Because all learned firm decisions occur at the beginning of the month, the learning problem is formulated at the monthly level. The RL time step t therefore corresponds to an individual simulated month.

To maintain the local-information assumption of the LBM, the learned firm observes only firm-level information comparable to the information available to a baseline firm. The decision problem is therefore partially observable from the perspective of the RL agent and is treated as an MDP approximation for training purposes. The challenge introduced by this partial observability and the use of a recurrent policy to address it are described in Section 3.3.1.

3.2.2 Observation and action spaces

At the beginning of each month, the learned firm receives an observation vector containing local firm-level information comparable to the information used by baseline

firms to adjust goods price and wage and manage workforce. Let o_t denote the observation the learned firm receives at month t; the observation vector is given by

$$o_t = \left(v_f,\ s_f,\ |B_f|,\ m_f^{\text{buf}},\ i_f,\ w_f,\ p_f,\ d_f\right),$$

where the variables on the right-hand side are described in Table 3. Consistent with the decentralized-information assumption of the LBM, the learned firm does not observe the full state of the economy, including aggregate market conditions, competitors' prices or wages.

After observing o_t, the learned firm takes an action a_t, defined as

$$a_t = \left(a_t^w, a_t^p, a_t^{\text{hr}}\right),$$

where a_t^w is the wage adjustment, a_t^p is the goods price adjustment and a_t^{hr} is the workforce action. Similar to the observation space, the wage and price adjustments are subject to the same restrictions imposed in the baseline firm's processes, i.e.

$$a_t^w \in [-\delta, \delta],$$
$$a_t^p \in [-\vartheta, \vartheta],$$

where δ and ϑ are the same maximum adjustment rates used in the baseline wage and price setting rules. The firm's wage and goods price at the beginning of month t are then updated according to

$$w_f \leftarrow w_f(1 + a_t^w),$$
$$p_f \leftarrow p_f(1 + a_t^p).$$

Additionally, to maintain the model's price-setting dynamics, the Calvo stickiness property is also applied to the learned firm's price adjustment: the new price is implemented only with probability θ_{calvo}; otherwise, the previous price is retained.

The workforce (human resource/HR) action is discrete,

$$a_t^{\text{hr}} \in \{0, 1, 2\},$$

where 0 denotes firing one worker, 1 denotes no workforce adjustment and 2 denotes opening one vacancy. If $a_t^{\text{hr}} = 0$ and the firm has at least one employee, one worker is selected at random for dismissal at the beginning of the next month, following the same firing rule as for the baseline firms. This action does not modify the firm's vacancy indicator. If $a_t^{\text{hr}} = 1$, neither the workforce nor the vacancy indicator is modified by

the HR action. If $a_t^{\text{hr}} = 2$, the firm's vacancy indicator is set to $v_f = 1$.

This action space design keeps the learned firm's decisions comparable to the baseline firm's decisions, limiting artificial advantages that would make the learned policy difficult to compare with the baseline firm behavior.

3.2.3 Reward function

Even with a clearly defined objective for the learned firm, the reward function used in this thesis went through several iterations. Before settling on the final version, preliminary reward functions gave rise to unexpected behaviors in the learned firm.

Notably, using monthly nominal profit directly as the reward function did not produce satisfactory behavior. A policy trained with this reward function learned very early on to repeatedly increase the goods price and decrease the wage until the firm went bankrupt (formally defined in Section 3.2.4) after losing customers and employees. While the learned firm achieved high short-term nominal profits, its short lifespan made it unsuitable for analyzing macroeconomic impacts.

Similarly, reward functions that did not account for longevity or survival tended to produce unstable policies that went bankrupt early compared with firms that generated sustainable profit. An intuitive way to interpret this issue is that a firm that survives for a long time is likely to generate more accumulated profit than a firm with high but unstable profit, as long as the sustainable firm is not too far behind in terms of monthly profitability.

With these considerations in mind, the reward function used in this thesis is based on the firm's real profit, defined as nominal profit divided by the consumer price index (CPI), with a fixed survival bonus for every month the firm remains alive. Let $R_{f,t}^{\text{sales}}$ denote the realized sales revenue of firm f during month t,

$$R_{f,t}^{\text{sales}} = \sum_{h=1}^{H} \sum_{k \in \mathcal{K}_{h,f,t}} p_f q_h^{(k)},$$

where $\mathcal{K}_{h,f,t}$ is the set of purchase attempts in month t in which household h buys goods from firm f, and $q_h^{(k)}$ is the quantity purchased in purchase attempt k. Let $C_{f,t}^{\text{labor}}$ denote the firm's labor costs during the same month. Nominal profit is then given by

$$\Pi_{f,t}^{\text{nom}} = R_{f,t}^{\text{sales}} - C_{f,t}^{\text{labor}}.$$

Let CPI_t denote the CPI of month t, calculated as a demand-weighted average goods

price using the demand encountered by each firm during that month,

$$\mathrm{CPI}_t = \frac{\sum_{f=1}^{F} p_f d_f}{\sum_{f=1}^{F} d_f},$$

where p_f and d_f are evaluated for month t. The reward received by the learned firm after month t is calculated according to

$$R_{t+1} = \mathrm{clip}\left(\frac{\Pi_{f,t}^{\mathrm{nom}}}{\kappa \mathrm{CPI}_t}, -5.0, 5.0\right) + \iota I_{t+1}^{\mathrm{alive}},$$

where $\kappa > 0$ is a profit scaling hyperparameter, $\iota > 0$ is the survival bonus hyperparameter, and I_{t+1}^{alive} is an indicator equal to 1 if the firm has not reached a terminal state and 0 otherwise. The values of κ and ι used in training are reported with the training hyperparameters in Section 3.3.3.

Notably, the CPI is used only in the reward calculation during training and is not included in the learned firm's observation vector, preserving the local-information assumption. The reward function clips the scaled real profit to $[-5.0, 5.0]$ as a training-stabilization technique for PPO. The survival bonus is included to discourage short-lived strategies that generate high immediate profits but quickly lead to bankruptcy. As a result, the learned firm is rewarded not only for generating profit, but also for remaining economically viable over time.

3.2.4 Episode structure and termination

The LBM is a continuing ABM at both the macro and micro levels. Without an artificial stopping condition, the simulated economy continues indefinitely. The number of firms is constant, and there is no endogenous firm entry or exit: an economically unviable firm is not replaced and remains in the economy until it becomes viable again. However, to meaningfully analyze the economic properties of an individual firm, a notion of firm lifecycle has to be specified. In this thesis, a firm is considered bankrupt, or economically unviable, when it has zero liquidity, zero liquidity buffer, zero inventory and zero employees.

The RL interaction is organized into episodes, where the learned firm repeatedly interacts with the simulated economy and one time step corresponds to one simulated month. An episode terminates when the learned firm is either bankrupt or no longer meaningfully viable as an economic unit, defined as experiencing 24 consecutive months of non-positive nominal profit. While bankruptcy serves as the primary

terminal condition of the episode, the non-positive-profit condition is a training-stabilization technique that shortens the RL agent's time spent in economically inactive or persistently unprofitable states.

3.3 Training setup and implementation

3.3.1 Partial observability and algorithm selection

Given the learning problem defined in Section 3.2, PPO is a suitable training algorithm for this thesis. PPO is a policy-gradient algorithm that trains a stochastic parameterized policy directly, making it compatible with the learned firm's mixed action space: the workforce action is discrete, while the wage and price adjustments are continuous. As an on-policy algorithm, PPO is less sample-efficient than many off-policy algorithms, but this limitation is less problematic in this thesis because interaction data can be generated on demand from the LBM simulation. Similarly, the availability of simulated interaction data reduces the need for an offline RL method. Aside from this technical suitability, PPO was also chosen because of its implementation simplicity and relative training stability compared with earlier policy-gradient algorithms [24].

However, as briefly mentioned in Section 3.2.1, the local-information assumption of the LBM makes the learned firm's decision problem partially observable. The learned firm does not observe the full state of the economy, but only a local firm-level observation vector. As a result, the current observation may not contain all information needed for effective decision-making: similar observations can correspond to different underlying economic states depending on unobserved variables and previous events. This problem can be addressed by introducing memory into the policy [12]. Instead of using a feedforward policy that maps only the current observation to an action, this thesis uses a recurrent policy that can condition its decisions on a history of observations [8]. The recurrent component used in this thesis is long short-term memory (LSTM) [9], a well-established and practical architecture that has also been used in other PPO-based RL systems [1, 20].

3.3.2 Software implementation

The software for this thesis was implemented in Python. The extended LBM was built with Mesa, an open-source ABM framework [10]. The custom RL environment follows the Gymnasium interface [27]. The recurrent PPO implementation was written in PyTorch [21] and adapted from CleanRL [11]. The training workflow and

experiments were conducted in Jupyter notebooks [13].

The policy and value functions share an initial feedforward feature network followed by an LSTM layer. The output of the LSTM layer is used by separate output heads: a value head for the critic, a categorical policy head for the discrete workforce action and a Gaussian policy head for the continuous wage and price adjustments. The policy samples the continuous actions in an unconstrained space, applies a tanh squashing function and rescales the outputs to ensure that the continuous actions remain within the bounds defined in Section 3.2.2. The log-probability used in the PPO objective is corrected for this transformation. Observation normalization is implemented inside the agent rather than the environment, using a running mean and variance estimator to support stable training with parallel environments.

The training code was run using the Metal Performance Shaders (MPS) backend on Apple Silicon. The reported training runs therefore reflect the MPS/PyTorch execution setup used in this thesis, although the choice of backend does not change the model formulation or environment dynamics.

3.3.3 Training procedure and hyperparameters

In addition to the terminal conditions described in Section 3.2.4, training episodes were truncated after 500 years, or 6,000 monthly steps. This truncation serves as a practical time limit to prevent episodes from continuing indefinitely when the learned firm remains viable for a long period. Without such a limit, exceptionally long episodes would reduce the frequency with which the agent is reset into other economic environments, thereby limiting the diversity of conditions encountered during training. Notably, evaluation instead uses a shorter standardized horizon of 100 years to keep repeated simulations computationally manageable while allowing sufficient evaluation samples to be collected.

Each extended LBM used in training was initialized with 99 baseline firms and one learned firm before being simulated for a burn-in period of 1,000 months, or approximately 83 years, similar to Lengnick [16]. The burn-in reduced the influence of the model's arbitrary initial conditions, and no evaluation statistics were collected during this period. During burn-in, the learned firm follows the same decision rules as the baseline firms to ensure that the post-burn-in state of the extended model remains consistent with that of the post-burn-in baseline model.

To reduce training time, the complete model state at the end of each burn-in period was saved as a post-burn-in snapshot. Training episodes begin from these

Table 5: Training hyperparameters used for recurrent PPO.

Hyperparameter	Value	Description
Training steps	2,000,000	Total number of monthly environment steps
Parallel environments	8	Number of asynchronous vectorized environments
Rollout length	1,024	Number of monthly steps collected per environment before each update
Batch size	8,192	Number of transitions per PPO update
Minibatches	4	Number of minibatches per update
Minibatch size	2,048	Number of transitions per minibatch
Update epochs	10	Number of optimization epochs per PPO update
Learning rate	0.0003	Initial learning rate of the optimizer
Discount rate γ	0.999	Discount rate applied to future rewards
GAE parameter λ_{GAE}	0.95	Bias-variance trade-off parameter for GAE
PPO clipping parameter ϵ	0.2	Clipping range for the PPO surrogate objective
Value-function coefficient c_V	0.5	Weight of the value function loss
Entropy coefficient c_H	0.0	Weight of the policy entropy bonus
Maximum gradient norm	0.5	Gradient clipping threshold
Advantage normalization	Yes	Advantages normalized within minibatches
Profit scale κ	10,000	Scaling parameter for real-profit reward
Survival bonus ι	0.01	Monthly reward bonus while the firm remains alive
Non-positive-profit threshold	24 months	Number of consecutive non-positive-profit months before termination
Episode time limit	6,000 months	Maximum episode length, equivalent to 500 simulated years

snapshots, avoiding the need to repeat the burn-in process before every episode. Twenty snapshots were generated using model seeds 0 through 19. At the start of each training episode, one snapshot was selected uniformly at random. Each model seed initialized the stochastic processes of the ABM, producing a distinct burn-in trajectory and post-burn-in economy.

Training was conducted using asynchronous vectorized environments to improve performance. The learning rate was annealed linearly over the course of training. The training hyperparameters are summarized in Table 5.

The pre-burn-in initial state variables used to generate these snapshots are summarized in Table 6.

Table 6: Initial state variables of firms and households.

Variable	Value
Initial firm goods price	40
Initial firm wage rate	2,000
Initial firm liquidity	0
Initial firm inventory	0
Initial household liquidity	5,000
Initial household reservation wage	0

3.4 Evaluation design and experimentation

After training, the learned policy was evaluated using 100-year simulations, corresponding to 1,200 monthly steps, from post-burn-in evaluation snapshots. The experiments provide two main assessments: first, whether the learned firm achieves the objective of profit maximization, and second, whether the learned behavior produces observable systemic effects on the economy.

3.4.1 Evaluation protocol

Following the snapshot-generation procedure described in Section 3.3.3, 20 evaluation snapshots were generated using model seeds 20 through 39. These snapshots were separate from the training snapshots generated using seeds 0 through 19. The evaluation models used the same parameters as the training models, except that the number of learned firms varied across experiments. As during training, all learned firms follow the baseline decision rules during the burn-in period. During the evaluation

simulations, statistics were collected at monthly intervals, as detailed in the following section.

The firm-level behavior and performance evaluation was conducted using models with 99 baseline firms and one learned firm. Evaluation runs were terminated when the learned firm became bankrupt or when the 100-year evaluation horizon was reached. If the learned firm was already bankrupt at the start of a post-burn-in snapshot, that evaluation seed was excluded from the firm-level evaluation. To evaluate the learned policy relative to the baseline heuristic, counterfactual simulations of a baseline firm starting from the same economic state were also conducted. Because the baseline heuristic is stochastic, each counterfactual simulation was repeated 20 times using baseline behavior seeds 100 through 119.

System-level effects were evaluated under four adoption scenarios, defined by the share of firms controlled by the learned policy: 0% adoption, with no learned firms; 1% adoption, with one learned firm; 50% adoption, with 50 learned firms; and 100% adoption, with 100 learned firms. In scenarios with multiple learned firms, all learned firms used the same trained policy. Each system-level simulation continued until the 100-year evaluation horizon was reached. Learned firms that became bankrupt during these simulations were not removed from the economy and continued to use the learned policy, consistent with the no-entry/no-exit structure of the LBM.

3.4.2 Outcome metrics and statistics

The evaluation uses firm-level and system-level outcome metrics calculated at monthly intervals. All quantities in the following metric definitions refer to the same monthly observation, so the month subscript is omitted for readability. A firm is classified as active if it does not satisfy the bankruptcy condition defined in Section 3.2.4. Let $\mathcal{F}^A$ denote the set of active firms and $\mathcal{F}^L \subseteq \mathcal{F}^A$ the set of active learned firms. Unless stated otherwise, market statistics are calculated using active firms only.

Firm-level outcomes Firm-level performance is evaluated in terms of longevity, profitability and operating behavior. Longevity is measured using survival length until bankruptcy and the full-horizon survival rate, defined as the share of evaluation runs in which the firm survives the complete 100-year evaluation horizon. Survival length is capped at 100 years.

Profitability is evaluated using cumulative nominal profit, mean monthly nominal profit and monthly profit percentile. Cumulative nominal profit is calculated by

summing monthly nominal profit over the firm's observed lifespan. The monthly profit percentile measures the learned firm's position in the distribution of nominal profits among active firms in the same economy. It is calculated as the percentage of active firms with lower profit, plus half the percentage with equal profit. The learned firm itself is included in this distribution whenever it is active. Values near 100 therefore indicate that the learned firm is among the most profitable active firms, whereas values near zero indicate that it is among the least profitable.

For direct comparisons between the learned policy and the baseline heuristic, selected statistics are calculated over a fixed horizon of 120 months. Mean monthly profit, employee count and customer count are calculated over this complete horizon. If a simulation terminates before month 120, the remaining observations for these variables are treated as zero. This represents the absence of subsequent profit and operating activity after the firm becomes bankrupt. Goods price and wage rate are instead averaged over the available observations within the first 120 months, without adding zero values after bankruptcy, because zero would not represent an economically meaningful price or wage.

Operating behavior is examined using employee count, customer count, inventory, liquidity buffer, goods price and wage rate. Customer count refers to the number of households whose supplier sets contain the firm. The learned firm's decisions are analyzed using wage and price adjustments, their absolute magnitudes and the frequencies of the three workforce actions: firing, no adjustment and hiring.

Additional measures used to interpret the learned firm's market position and operating efficiency, including relative price and wage differences, profit margin and profit per employee, are defined at their first use.

System-level outcomes System-level outcomes are divided into labor market outcomes, goods market outcomes and market structure statistics. Aggregate quantities are calculated by summing over active firms, while firm-level means and medians are calculated across active firms in each month.

The main labor market and goods market metrics are defined in Table 7.

Real consumption is calculated by deflating aggregate realized sales revenue using the demand-weighted CPI from Section 3.2.3, here calculated across active firms.

The consumption-production gap is positive when realized real consumption exceeds current production and negative when production exceeds realized consumption. The inventory-to-demand ratio is calculated as the ratio of aggregate inventories to aggregate encountered demand, rather than as the mean of firm-level

inventory-to-demand ratios.

Table 7: Definitions of the labor market and goods market outcome metrics.

Metric	Calculation
Unemployment rate (%)	$100\left(1-\frac{\sum_{f\in\mathcal{F}^A}\lvert B_f\rvert}{H}\right)$
Mean wage rate of active firms	Arithmetic mean of w_f across $f\in\mathcal{F}^A$
Median wage rate of active firms	Median of w_f across $f\in\mathcal{F}^A$
Aggregate production	$\sum_{f\in\mathcal{F}^A} 21\lambda\lvert B_f\rvert$
Real consumption	$\frac{\sum_{f\in\mathcal{F}^A} R_f^{\text{sales}}}{\text{CPI}}$
Consumption-production gap (%)	$100\frac{\text{real consumption} - \text{aggregate production}}{\text{aggregate production}}$
Mean household unsatisfied demand ratio (%)	Mean household-level unsatisfied demand percentage defined below
Inventory-to-demand ratio	$\frac{\sum_{f\in\mathcal{F}^A} i_f}{\sum_{f\in\mathcal{F}^A} d_f}$
Median goods price	Median of p_f across $f\in\mathcal{F}^A$

Household unsatisfied demand is accumulated during the daily purchasing process. After a household has completed its purchasing attempts on a given day, any remaining effective demand is added to its monthly unsatisfied demand total. Effective demand may already have been reduced by the household's liquidity constraint; consequently, consumption removed because the household lacks sufficient liquidity is not counted as unsatisfied demand.

At the end of the month, the household's accumulated unsatisfied demand is divided by its planned monthly real consumption. A household with zero planned consumption is assigned an unsatisfied demand ratio of zero. The reported monthly statistic is the arithmetic mean of these ratios across all households, multiplied by 100. This procedure gives equal weight to each household rather than calculating the ratio of aggregate unsatisfied demand to aggregate planned consumption.

The level and volatility of unemployment are summarized using the mean and standard deviation of the monthly unemployment rate within each simulation run. Wage conditions are summarized using the means over months of the monthly mean and median active firm wage rates. Goods price conditions are summarized using the mean and standard deviation over months of the monthly median goods price.

Market structure and profitability The market structure and profitability measures reported in the evaluation are defined in Table 8.

Table 8: Definitions of the market structure and profitability metrics.

Metric	Calculation
Active firm count	$\lvert\mathcal{F}^A\rvert$
Active learned firm count	$\lvert\mathcal{F}^L\rvert$
Active baseline firm count	$\lvert\mathcal{F}^A\rvert - \lvert\mathcal{F}^L\rvert$
Revenue share of learned firms (%)	$100\dfrac{\sum_{f\in\mathcal{F}^L} R_f^{\text{sales}}}{\sum_{f\in\mathcal{F}^A} R_f^{\text{sales}}}$
Revenue Herfindahl–Hirschman index (HHI)	$\displaystyle\sum_{f\in\mathcal{F}^A}\left(\frac{R_f^{\text{sales}}}{\sum_{g\in\mathcal{F}^A} R_g^{\text{sales}}}\right)^2$
Aggregate nominal profit	$\displaystyle\sum_{f\in\mathcal{F}^A}\Pi_f^{\text{nom}}$
Aggregate profit of learned firms	$\displaystyle\sum_{f\in\mathcal{F}^L}\Pi_f^{\text{nom}}$
Profit per active firm	$\dfrac{\sum_{f\in\mathcal{F}^A}\Pi_f^{\text{nom}}}{\lvert\mathcal{F}^A\rvert}$
Profit per active learned firm	$\dfrac{\sum_{f\in\mathcal{F}^L}\Pi_f^{\text{nom}}}{\lvert\mathcal{F}^L\rvert}$

Revenue HHI is the unnormalized sum of squared revenue shares of active firms; higher values indicate greater revenue concentration.

Statistical aggregation Unless stated otherwise, monthly statistics were first summarized within each simulation run, and the resulting run-level values were then averaged across evaluation seeds. Means were calculated across months within each run, while standard deviations measured variation across months within each run. Ratios were calculated monthly before averaging rather than from quantities that had already been averaged.

For analysis of the baseline firm's performance, the repeated counterfactual simulations within each evaluation seed were first summarized using their median. The resulting seed-level values were then averaged across evaluation seeds. The exception was the full-horizon survival rate of baseline firms, which was calculated as the share of all repeated counterfactual simulations that survive the complete 100-year horizon.

Calculations with a zero denominator are treated as undefined and excluded from summaries requiring valid monthly observations. Unless otherwise specified, proportions are multiplied by 100 when reported as percentages. Additional descriptive measures and trajectory-specific analysis procedures used only for particular results are defined at their first use in the subsequent analysis.

The source code, trained policy, post-burn-in model snapshots, simulation outputs, and notebooks used to produce the results are publicly available through Zenodo [15].

4 Results

The learned policy was evaluated in 20 runs initialized from independently generated post-burn-in model states corresponding to model seeds 20-39. For four model seeds, the resulting post-burn-in state contained a learned firm that was effectively bankrupt at the start of evaluation. These runs were therefore excluded from the firm-level analysis but retained in the system-level analysis. The remaining 16 model seeds, whose post-burn-in states contained an economically active learned firm, are referred to as viable evaluation seeds. In the corresponding runs, the learned firm survived the full simulation horizon in 15 and failed to do so in one.

4.1 Learned firm behavior

The learned firm's behavior is characterized by a transition from an unstable pre-transient period, in which the firm exhibits relatively large and fluctuating adjustments, to a more stable post-transient regime, in which adjustments become smaller and profitability remains persistently high relative to other active firms. Reaching this stable regime is common among the evaluation seeds, but it is not guaranteed: in one viable evaluation seed, the firm failed to reach the high-performance regime. The transition is also path-dependent, as the time required to reach this regime varies across seeds depending on the firm's starting condition.

4.1.1 Representative transition dynamics

Figure 3 illustrates an example simulated trajectory in which the learned firm reaches a relatively stable post-transient operating regime. The pre-transient period is characterized by repeated drops in the liquidity buffer, temporary inventory accumulation and relatively large wage and price adjustments. During this phase, the learned firm's profitability is unstable, as indicated by the large swings in profit percentile. At the same time, the gradual increase in employee count and customer count suggests that the firm is expanding despite this instability.

After about 150 months, these variables become markedly more stable. The learned firm rises to the top profit percentile and remains there for most of the remaining simulation horizon. Inventories transition to a persistently low level, while the number of Type A connections increases substantially, approaching the size of the household population. The adjustments the learned firm makes to wages and goods prices stabilize and remain relatively smaller than in the initial phase.

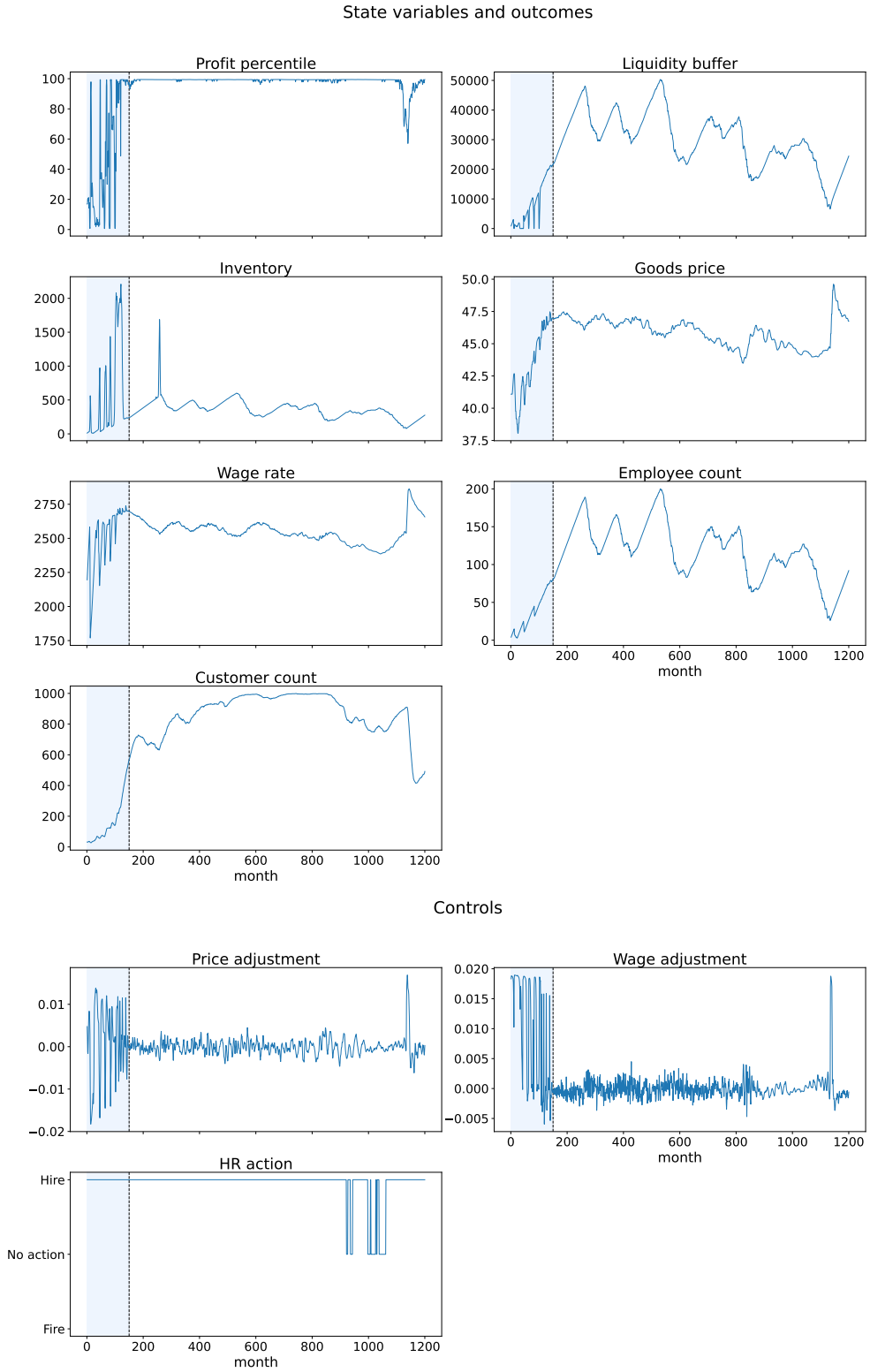

Figure 3: A representative monthly trajectory of the learned firm under baseline conditions (seed 36). The blue shaded area marks the pre-transient period for this seed (months < 150), and the dashed vertical line marks the corresponding transition point.

Notably, although the post-transient regime is substantially more stable than the pre-transient period, it is not completely static: fluctuations in customer count, employee count, and profit percentile remain visible later in the simulation. The HR action taken by the learned firm also remains heavily concentrated in the hiring option, with only occasional deviations. As will be shown later for other seeds, this strong preference for the hiring action is common across the evaluation runs, suggesting that the HR branch is behaviorally more concentrated than the continuous control dimensions.

Figure 4: Comparison of representative transition patterns across selected seeds. Differences between an early-transition trajectory (seed 20), a late-transition trajectory (seed 22) and a trajectory with no identified transition (seed 30) are highlighted using low-noise state variables and outcomes. The blue and orange dashed vertical lines mark the visually identified transition points for the early- and late-transition runs, respectively.

While Figure 3 examines one transition in detail, Figure 4 highlights variation in transition timing and outcome by comparing three additional representative trajectories: an early-transition run, a late-transition run and a run with no identified transition. Similar to the full trajectory shown in Figure 3, the early-transition trajectory in Figure 4 reaches the high-performance regime relatively quickly, at around the 150-month

mark. However, in this case, the learned firm remains stable for the rest of the simulation horizon, with little subsequent variation in profit percentile or liquidity buffer.

The late-transition case, while exhibiting the same general transition and post-transient operating pattern, differs from the previous representative trajectories in that its transition point occurs at the 400-month mark; relative to the early-transition case, the learned firm requires more than twice as long to reach the high-performance regime. By contrast, the case with no identified transition never reaches the stable regime; the learned firm remains trapped in the pre-transient phase until it eventually goes bankrupt.

Taken together, the trajectories in Figures 3 and 4 that reach the stable post-transient regime suggest that the learned policy is associated with a common high-performance operating pattern. However, whether and how quickly the firm reaches that regime varies across seeds. Depending on the firm's starting condition and the surrounding market conditions, the learned firm may transition rapidly, transition only after a prolonged transient period, or fail to reach the regime altogether.

In the analysis above, the transition point is used to denote the approximate beginning of the more stable operating regime. For each seed, this point was identified visually based primarily on the transition to persistently high profit percentile, with the trajectories of wage and price adjustments as secondary indicators. Therefore, the transition point should be interpreted only as an approximation and not the exact month at which all variables stabilize.

4.1.2 Comparison with baseline firms

Due to the stochastic nature of the baseline heuristics, different simulations of the baseline firm from the same starting state can lead to different trajectories and, more importantly, different survival periods. This variation in baseline lifespan makes one-to-one counterfactual comparisons against the learned firm sensitive to the particular baseline seed chosen. To address this, two complementary comparisons are presented: one against the median active-firm population in the same simulated economy, and one against an illustrative baseline counterfactual trajectory from a post-burn-in state that survives the longest among the evaluation seeds.

Figure 5 compares the learned firm with the monthly active firm median for each reported variable. Relative to the median active firm, the learned firm maintains a substantially larger operating scale, as indicated by its much larger customer base and

workforce. At the same time, the learned firm's goods price remains close to, but generally somewhat below, that of the median active firm, while its wage rate remains somewhat higher. As will be discussed in more detail in Section 5.1, one common descriptive behavior of the learned firm across the evaluation seeds is that the learned firm tends to prefer a slightly but persistently lower goods price and a higher wage rate than the typical active firm.

Figure 5: Comparison of the learned firm with the monthly medians of goods price, wage rate, employee count, and customer count across all active firms in the same simulated model (seed 34). Relative to the median active firm, the learned firm operates at a substantially larger scale, while maintaining slightly lower prices and slightly higher wages.

Figure 6 compares the learned firm with a simulated trajectory for the baseline heuristics starting from the same post-burn-in condition. This baseline trajectory is simulated counterfactually, as if the firm had continued to follow the original heuristic rules rather than adopting the learned policy. To make the qualitative comparison both meaningful and reproducible, the figure uses the seed that produces the longest-surviving baseline trajectory among the evaluation seeds.

Similar to the comparison in Figure 5, the learned firm again maintains a substantially larger operating scale than the baseline firm. In addition, the learned firm appears to be more stable financially, indicated by its large post-transient liquidity buffer and the absence of the noticeable wage cuts visible in the baseline trajectory. While the median active firm comparison in Figure 5 emphasizes the learned firm's position relative to the surviving market population, Figure 6 shows that even in a favorable baseline trajectory the baseline firm remains much smaller and financially

weaker than the learned firm.

Notably, the baseline firm often sets lower prices and pays lower wages than the learned firm, yet this does not translate into comparable operating scale. The figure therefore suggests that the difference between the two firms is not simply a matter of lower prices or lower labor costs, but of a broader difference in the operating regime that each firm is able to sustain.

Figure 6: Illustrative trajectory comparison between a learned firm and a baseline firm simulated from the same post-burn-in initial state (seed 34). The seed is chosen because it produced the longest-surviving baseline trajectory among the evaluation seeds and therefore provides the most informative qualitative comparison.

4.1.3 Cross-seed behavioral summary

Table 9 summarizes the differences between the pre-transient period and the post-transient regime using mean statistics of the learned firms across the evaluation seeds. Similar to the qualitative analysis in Section 4.1.1, transition points were identified visually for each seed using primarily profit percentile and secondarily the trajectories of price and wage adjustments. Different transition cutoffs were required because the identified transition points ranged from 75 to 400 months across seeds with an identified transition, with a cross-seed standard deviation of 80 months, making a

single common cutoff unsuitable for this analysis. Pre-transient and post-transient means were first calculated within each seed before being averaged across seeds for the summary table. The statistics of the seed with no identified transition are used to calculate the pre-transient means only.

Consistent with the representative trajectory analysis in the previous section, the cross-seed summary shows clear differences between the two phases. Relative to the pre-transient phase, the post-transient operating regime is characterized by markedly greater stability and larger operating scale. While some of the differences in profit percentile and adjustment magnitudes can be attributed to the way the transition points were identified, the learned firm's consistent placement near the top of the profit ranking after transition suggests that a stable high-performing operating regime exists in the baseline model. The reduction in absolute price and wage adjustments across seeds matches the earlier qualitative observation that the learned firm's behavior becomes less volatile after the transition. Likewise, the large increases in workforce and customer base suggest that, in addition to being more stable, the post-transient regime is also associated with a substantially larger operating scale.

Table 9: Mean pre-transient and post-transient statistics across 16 viable evaluation seeds. Post-transient statistics are computed only for seeds with an identified transition.

Statistic	Pre-transient mean	Post-transient mean
Profit percentile	58.63	99.03
Absolute price adjustment	0.49%	0.12%
Absolute wage adjustment	0.75%	0.11%
Employee count	49.28	146.41
Customer count	189.49	844.68
Liquidity buffer	11,029.29	37,602.59

Contrary to the more nuanced behavior observed in wage and price adjustments, the HR control actions chosen by the learned firm are notably one-sided. Table 10 shows that in both the pre-transient and post-transient phases, the learned firm chooses almost exclusively to hire. Across all evaluation seeds, no firing actions were recorded, and the no-action choice appears only rarely after transition. This cross-seed result confirms the earlier observation from the representative trajectories that the HR branch is behaviorally much more concentrated than the continuous price and wage controls.

Table 10: Distribution of HR actions across 16 viable evaluation seeds. The seed with no identified transition contributes only to the pre-transient distribution.

Phase	Fire share	No action share	Hire share
Pre-transient	0.00%	0.00%	100.00%
Post-transient	0.00%	0.67%	99.33%

4.2 Firm-level outcomes

While the previous section examines the patterns of the learned firm's behavior, this section evaluates whether this behavior leads to higher performance in profit generation relative to the baseline heuristic. Following the evaluation protocol described in Section 3.4.1, the learned policy is compared against counterfactual baseline simulations initialized from the same post-burn-in economy states.

Because baseline firms typically go out of business much earlier than learned firms, selected statistics are computed only over a fixed first-10-year horizon to improve comparability. These statistics should therefore be interpreted as early-horizon estimates rather than full-horizon summaries of the learned policy's performance; in some evaluation seeds, the visually identified transition points occur after the 120-month mark.

4.2.1 Survival and financial outcomes

Table 11 compares the learned firm and baseline firm in terms of survival and financial metrics. The learned firm substantially outperforms the baseline firm in all measured performance outcomes. In 15 out of the 16 simulations, the learned firm reaches the end of the 100-year simulation horizon, while the baseline firm reaches the time limit in only 2 out of the 320 stochastic baseline simulations. The mean observed lifespan of the learned firm is also more than six times as long as the baseline lifespan. Since survival length is capped at 100 years, the reported difference may understate the full difference in expected longevity.

The difference in survival outcomes translates into a large disparity in cumulative profit. On average, the learned firm makes close to 50 times more cumulative profit than the baseline firm over their observed lifespans. Notably, this difference in lifetime profit generation is much larger than the difference in survival length alone, suggesting that, in addition to increased longevity, the learned policy also improves firm-level profit outcomes.

Cumulative profit, however, is not a fully equal one-to-one comparison for

Table 11: Comparison of firm-level survival and financial outcomes between the learned policy and the baseline heuristic across viable evaluation seeds.

Outcome	Learned policy	Baseline heuristic
Full-horizon survival rate (%)	93.75%	0.63%
Survival length (years, capped at 100)	95.48	14.56
Cumulative profit	53,405,015	1,093,510
Mean monthly profit (first 10 years)	11,539	4,909

evaluating profitability, because the statistics are recorded over different time periods. For that reason, Table 11 also reports mean monthly profit over the first 10 years, with baseline firms that go bankrupt before the 10-year mark counted as having zero monthly profit after exit. Even within this fixed horizon, the learned firm achieves more than twice as much monthly profit on average, indicating that the learned policy improves profit outcomes even during the early-horizon period, before all learned runs have necessarily reached the post-transient regime.

4.2.2 Scale and operating statistics

Table 12 reports and compares key operating statistics of the learned firm and baseline firm over the first 10 years of simulation. Consistent with the pattern observed in the behavioral analysis of Section 4.1, the learned firm operates at a substantially larger scale than the baseline firm, even during this early-horizon period. On average, the learned firm employs about five times as many workers as the baseline firm, while maintaining a customer base roughly three times as large. Importantly, this difference in scale is not due to the baseline firm being unusually small, since the baseline values are close to the approximation implied by the model's parameters under an even distribution of workers and supplier connections: with 1,000 households and 100 firms, an average firm would have about 10 employees, and with each household having 7 suppliers, the average firm would maintain about 70 customer connections.

The learned firm's larger scale is accompanied by a slightly lower mean goods price and a higher mean wage relative to the baseline heuristic. Over the first 10 years, the learned firm pays about 10% higher wages and charges around 1% lower goods prices than the baseline firm. The magnitude of these differences is small relative to the large differences in employee and customer counts, suggesting that the learned firm's outcome advantage is not simply explained by extreme price or wage deviations.

Table 12: Comparison of firm-level scale and operating statistics between the learned policy and the baseline heuristic across viable evaluation seeds.

Statistic	Learned policy	Baseline heuristic
Mean employee count (first 10 years)	50.11	10.11
Mean customer count (first 10 years)	205.23	67.93
Mean goods price (first 10 years)	44.80	45.31
Mean wage rate (first 10 years)	2,572.77	2,332.50

4.3 Macroeconomic outcomes

This section examines whether the firm-level performance improvements in the previous section lead to observable macroeconomic effects. The different adoption scenarios serve as counterfactual experiments to evaluate whether these consequences generalize across different adoption levels. Importantly, the policy used in all scenarios is the same policy trained in the setting with one learned firm, and not a newly optimized policy trained in a multi-agent setting. As such, the 50% and 100% adoption scenarios do not represent equilibria in which firms adapt to the behavior of other learned firms, but rather scaled implementations of a locally learned profit-oriented policy. This distinction is important because a policy that improves the outcome of one firm can produce different aggregate effects when many firms follow it simultaneously.

4.3.1 Labor market outcomes

Table 13 summarizes the outcomes of the labor market across different adoption scenarios. The simulation results show that the adoption of the learned policy is associated with both lower unemployment and lower unemployment volatility. Introducing one learned firm to the baseline economy cuts the unemployment rate by roughly 40%, and full adoption practically reduces unemployment to zero. The standard deviation of unemployment follows a similar trend, suggesting that the labor market becomes more stable as adoption increases.

In addition to lower unemployment, Table 13 also shows that the wage level of active firms increases as more firms adopt the learned policy. This trend can be observed in more detail in Figure 7, where a representative trajectory of the labor market outcomes is displayed. The wage trajectory shows a shift toward higher wages, with the 50% and 100% adoption scenarios also showing substantially more stable wage dynamics than the baseline scenario. Combined with the unemployment statistics, this suggests that the adoption of the learned policy leads to a tighter labor

Table 13: Labor market outcomes across adoption scenarios.

Statistic	0% adoption	1% adoption	50% adoption	100% adoption
Mean unemployment rate (%)	0.54	0.31	0.07	0.00
Std. unemployment rate (percentage points)	0.45	0.27	0.09	0.01
Mean wage rate of active firms	2,286.14	2,437.42	2,675.30	2,763.59
Mean median wage rate of active firms	2,329.34	2,485.43	2,773.71	2,786.43

market, characterized by lower unemployment and higher wages.

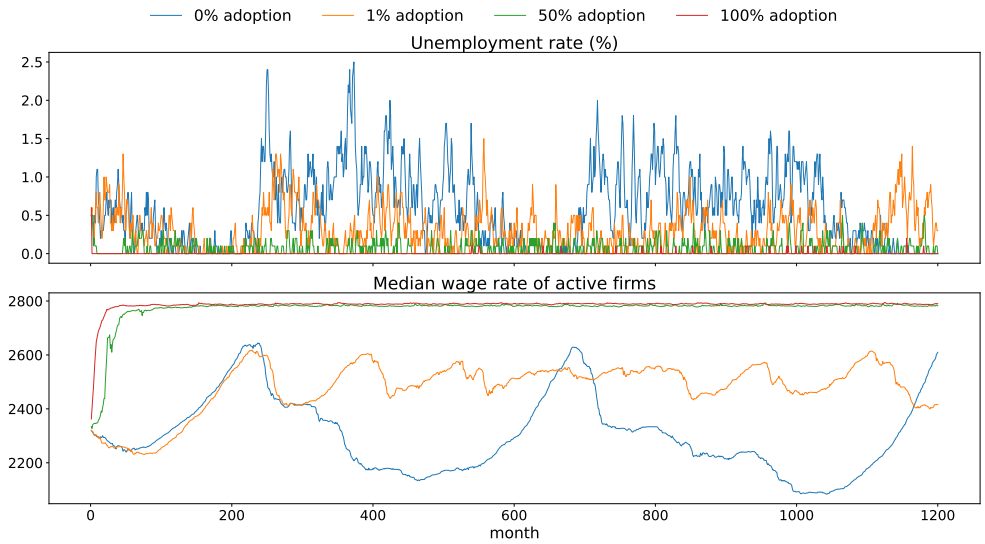

Figure 7: A representative monthly trajectory of the labor market under different adoption scenarios (model seed 20, run seed 100). Consistent with the cross-run summary in Table 13, higher adoption rates are associated with lower and less volatile unemployment, as well as higher and more stable wage levels among active firms.

4.3.2 Goods market outcomes

Table 14 summarizes the key simulation outcomes of the goods market. The results suggest that the adoption of the learned policy improves goods market coordination under partial adoption (1% and 50% adoption scenarios), but disrupts the balance between supply and demand under full adoption. With one learned firm in the economy, the gap between realized real consumption and production is reduced by

around 60% relative to the baseline, while unsatisfied household demand falls by a similar amount. The 50% adoption scenario exhibits the most coordinated goods market outcomes: realized real consumption and production are almost identical, and unsatisfied demand falls to about one-fifth of the baseline value.

Table 14: Goods market outcomes across adoption scenarios.

Statistic	0% adoption	1% adoption	50% adoption	100% adoption
Mean aggregate production	62,660	62,807	62,957	62,999
Mean real consumption (revenue/CPI)	62,811	62,869	62,958	62,934
Mean real consumption-production gap (% of production)	0.24	0.10	0.00	-0.10
Mean household unsatisfied demand ratio (%)	5.31	2.12	1.01	1.30
Mean inventory-to-demand ratio	0.09	0.11	0.09	0.44
Mean median goods price	45.00	46.74	46.80	46.66
Std. median goods price	2.05	1.04	0.32	0.26

However, the mean inventory-to-demand ratio indicates a disrupted balance between supply and demand in the 100% adoption scenario, where the ratio is roughly four to five times larger than in the other scenarios. This oversupply issue is further examined in Figure 8, where the illustrative trajectory shows that the inventory buildup is persistent under full adoption. While the 100% adoption scenario still improves both the real consumption-production gap and unsatisfied household demand relative to the baseline, these improvements come at the cost of goods market disequilibrium with accumulating excess supply.

In addition to the coordination outcomes, there are visible signs suggesting that the adoption of the learned policy is associated with more stable goods-price dynamics. Table 14 reports around 50% lower volatility in the market goods price (median price of active firms) in the 1% adoption scenario. This pattern can be seen more clearly in the representative price trajectory in Figure 8. Introducing even one learned firm into the economy raises the market price level by around 4%, although further increases in adoption do not appear to raise this price level further. This observation and its interpretation are discussed in more detail in Section 5.2.

Unlike the labor market outcomes, the statistics in the goods market do not move

monotonically with adoption. The 100% adoption scenario performs worse than the 50% adoption scenario in several goods market outcomes, with lower real consumption, higher unsatisfied demand and persistent excess inventories. This nonlinearity suggests that the systemic effect of the learned policy depends on the level of adoption: a policy that improves coordination of the goods market when adopted partially can produce overproduction when generalized to all firms.

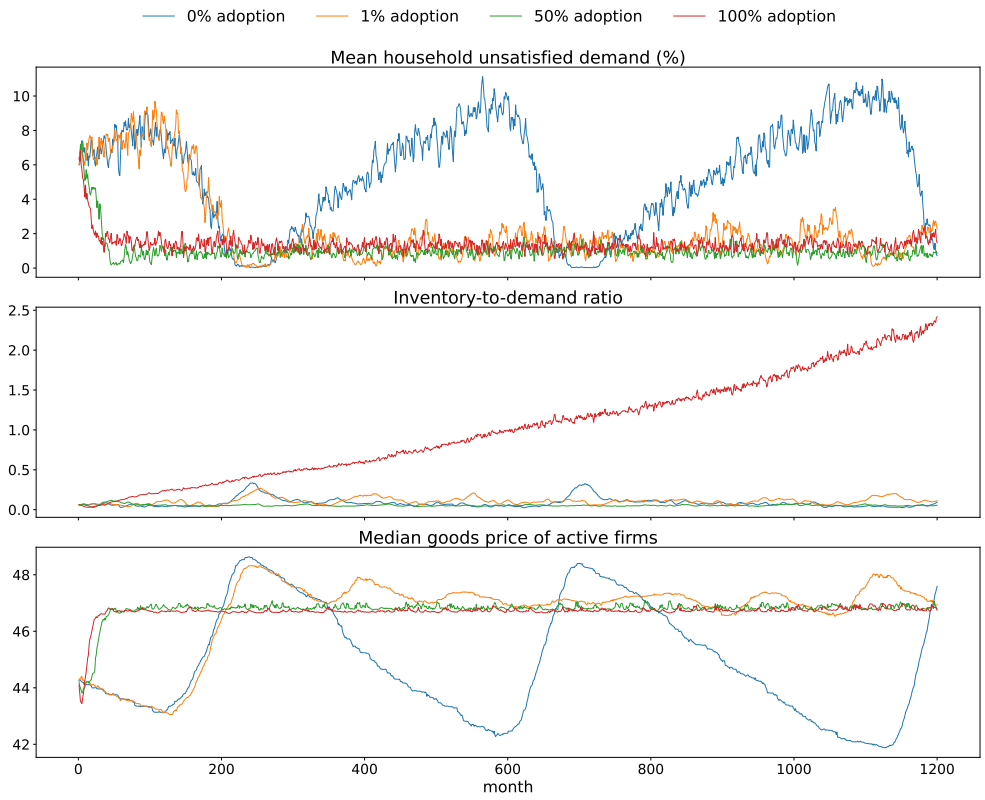

Figure 8: Representative monthly trajectories of the goods market under different adoption scenarios (model seed 20, run seed 100). All adoption scenarios exhibit lower unsatisfied demand than the baseline scenario, while full adoption produces persistent inventory accumulation.

4.3.3 Firm population, market concentration and profitability distribution

Although these statistics are not usually classified as macroeconomic outcomes, inspecting market structure and firm population statistics helps explain both the firm-level and system-level effects of learned policy adoption. Table 15 summarizes the statistics on firm population, market concentration and profitability distribution. Overall, the results illustrate both the dominance of the learned policy over the baseline heuristic and the changing distribution of market activity as adoption increases.

Table 15: Firm population, market concentration and profitability distribution across adoption scenarios.

Statistic	0% adoption	1% adoption	50% adoption	100% adoption
Mean active firm count	81.98	81.27	61.62	95.90
Mean active learned firm count	–	0.99	49.51	95.90
Mean active baseline firm count	81.98	80.27	12.11	–
Market share of learned firms (revenue) (%)	–	13.83	97.55	100.00
Revenue HHI	0.019	0.037	0.020	0.012
Mean aggregate profit	489,226	436,227	175,481	160,576
Mean learned firm aggregate profit	–	49,418	162,595	160,576
Mean profit per active firm	5,989	5,384	2,854	1,679
Mean profit per active learned firm	–	49,630	3,285	1,679

Consistent with the firm-level scale statistics, in the 1% adoption scenario the single learned firm becomes substantially larger than a typical baseline firm, capturing around 14% of market revenue while competing against around 80 active baseline firms. As a result, the 1% adoption market is much more concentrated, with the revenue HHI approximately doubling relative to the baseline market.

The dominance of the learned policy manifests in a different form in the 50% adoption scenario. While the market share dominance of the learned firm in the 1% adoption scenario does not substantially affect the active baseline firm population, in the 50% adoption economy around 75% of baseline firms go bankrupt on average. The remaining surviving baseline firms account for only around 2.5% of economic activity, despite baseline firms making up half of the initial firm population. The representative trajectory in Figure 9 illustrates that this dominance is persistent rather than only a temporary transition effect.

The concentration and profitability statistics further clarify how the effect of the learned policy changes with adoption. While one learned firm increases market concentration, higher adoption levels are associated with lower revenue HHI, suggesting that market activity becomes more evenly distributed across many learned firms rather than concentrated in a single large learned firm. The learned firm is highly profitable

when it is rare: in the 1% adoption scenario, the mean monthly profit of the learned firm is almost ten times higher than the average profit per active firm. However, mean monthly profit per active learned firm falls sharply from 49,630 in the 1% adoption scenario to 3,285 under 50% adoption and 1,679 under full adoption. This drop in profitability implies that the financial performance gain of the learned policy is partly positional: it is strongest when the learned firm competes against baseline firms, but weaker when learned firms increasingly compete with each other.

Figure 9: Representative monthly trajectories of firm population, learned firm revenue share and aggregate profit under different adoption scenarios (model seed 20, run seed 100). The adoption of the learned policy changes the composition of the active firm population, with partial adoption producing persistent dominance of the learned firms and higher adoption levels associated with lower aggregate firm profitability.

In addition to lower firm-level profitability, higher adoption levels are also associated with lower system-level aggregate profitability. This suggests that higher adoption shifts the economy toward lower-margin firm operation: the outcomes in the labor and goods markets improve in some dimensions, but these improvements are accompanied by lower aggregate profitability and a weaker individual profitability advantage for learned firms.

5 Discussion

5.1 The learned strategy

To better understand the performance gain of the learned policy, further inspection into how the learned firm acts is provided. While Section 4.1 gives an overview description of the learned firm's mechanical behavior, this section reviews and organizes the relevant statistics to form an interpretation of the strategy that the learned firm employs to dominate the 1% adoption market.

Here, relative price and wage differences are calculated as the learned firm's percentage deviations from the monthly median across active firms, with negative and positive values indicating values below and above the median, respectively. Profit margin is nominal profit divided by realized sales revenue, while profit per employee is nominal profit divided by employee count. These measures are calculated monthly before being summarized across time and evaluation seeds.

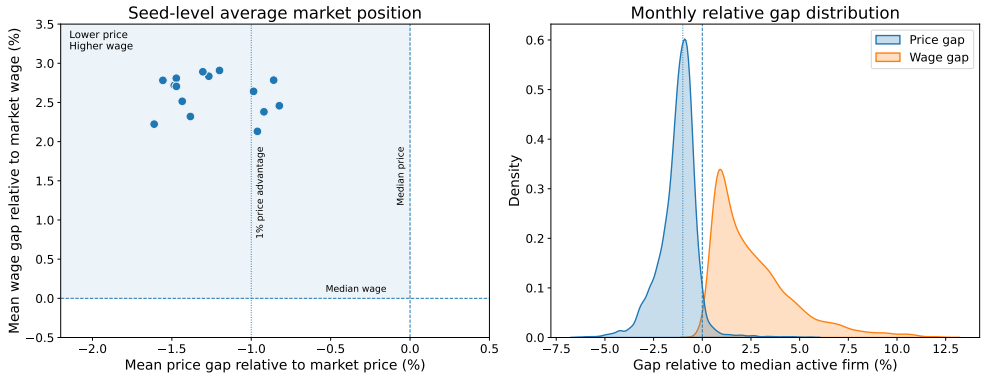

Figure 10: Relative positioning of the learned firm in the goods and labor markets. The left panel shows the seed-level mean price and wage gaps relative to the median active firm. The right panel shows the monthly distribution of these relative gaps across evaluation seeds. The dashed lines mark parity with the median active firm, and the dotted vertical line marks a 1% price advantage.

The results suggest that instead of a simple high-price or low-wage strategy, the learned firm achieves superior performance through a scale-expansion strategy that offers attractive terms in both the goods and labor markets. Across evaluation seeds, the learned firm consistently sets its goods price below the market price, while paying above the market wage: in 96.1% of evaluated months, the learned firm's goods price is below the median price of active firms, and in 99.8% of evaluated months, its wage is higher than the market median. On average, the learned firm's price is 1.25% lower than the market price, while it pays 2.61% higher wage than the market

wage. The learned firm's relative positioning is illustrated in more detail in Figure 10, highlighting the consistency of this pattern both across evaluation seeds and throughout the simulation periods.

It is possible to speculate on why this strategy would work well in the LBM. In a goods market with price-conscious customers and no differentiation between goods, lowering price is the primary way to gain a competitive advantage over other suppliers. Households attempt every month to improve their trading connections using the 1% price rule (parameter ξ in Table 4), and Figure 10 suggests that the learned policy frequently positions the firm close to this behaviorally relevant price gap, although this does not prove that the policy directly targets the switching threshold. Similarly, in order to gain enough production output to support a larger customer base, the learned firm needs to hire more workers, which it does by paying higher wages and essentially never firing any employee (Table 10 of Section 4.1.3). Moreover, having a larger workforce is also synergistic with lower prices in the LBM: in the price-based search, households have a higher probability of switching to suppliers with larger sizes, measured by employee count.

The distinction between the pre-transient and post-transient phases can also be interpreted through this scale-expansion logic. In the pre-transient phase, the learned firm is still moving from its initial state toward its preferred operating regime. During this period, the learned firm has not yet stabilized the relationship between realized demand and production capacity, because households and workers switch suppliers or employers gradually rather than immediately. After the transition, the firm operates at a more stable scale, with smaller price and wage adjustments and more controlled inventories.

Notably, the learned firm is capable of implementing this strategy despite not having access to aggregate market information, such as market price, competitors' prices or competitors' wages. While the learned strategy cannot be fully interpreted as a direct reaction to market-level variables, the consistency of the outcome suggests that the recurrent policy may infer relevant market conditions indirectly from local feedback signals.

There is, however, a clear tradeoff to this focus on scale. In exchange for higher total nominal profits, the learned firm accepts a lower profit margin: its post-transient margin is 12.17%, compared with the market margin of 14.64%. On average, the learned firm's profit per employee is around 18% lower than the market statistic. These results qualify the scale-expansion interpretation: the learned firm does not dominate because every performance dimension improves, but because its larger customer base

and workforce are sufficient to compensate for lower margins. This also helps explain why learned firms do not perform as well individually in the 50% and 100% adoption markets: when many firms follow a similar low-margin, scale-oriented policy, the positional advantage of the learned strategy is reduced.

The learned high-volume, lower-margin strategy also clarifies what kind of profit optimization the policy learns. Although the reward function is based on nominal profit, the successful behavior is not simply short-term profit extraction; instead, the learned policy appears to maximize sustainable profit by maintaining the conditions required for long-run survival and scale: sufficient demand, enough workers, controlled inventories and stable liquidity. In this sense, the learned firm trades off per-unit profitability for a larger and more stable operating scale.

5.2 Emergent macro effects

ABMs are most informative when they allow modelers to observe and study unexpected and counterintuitive system-level patterns. In the extended LBM, two noteworthy macroeconomic outcomes do not follow mechanically from the learned firm's local lower-margin, scale-expansion behavior. This section presents these observations and speculates about the possible mechanisms that could explain these patterns.

Lower relative prices can coexist with higher aggregate prices One initially counterintuitive result of the learned policy is the rise in the market price level when a learned firm is introduced into the baseline economy, as shown in Figure 8. Since the learned firm tends to price below the median active firm, one might expect its presence to put downward pressure on competitors' prices and therefore lead to an overall reduced market price level. Instead, the market price stabilizes at a higher level in the 1% adoption scenario than in the baseline economy.

This result becomes more understandable once the wage side of the learned strategy is considered. The learned strategy includes not only lower relative pricing, but also higher relative wage setting. The learned firm's higher relative wage offer, combined with its persistent hiring behavior, can exert upward pressure on other firms' wages, consequently leading to a higher overall market wage level and, as a result, a higher market marginal cost. The market price outcome therefore suggests that, in this setting, the upward pressure generated by higher wages and costs dominates the downward pressure generated by lower relative pricing, shifting the economy toward a higher-wage, higher-cost and higher-price regime.

However, the fact that market prices do not continue rising from 1% to 50% and 100% adoption suggests that this outcome is not just the result of a simple monotonic wage pass-through effect. Wages and implied marginal costs continue to increase at higher adoption levels, but median prices remain close to the 1% adoption level. This result suggests that additional cost pressure in the 50% and 100% adoption scenarios is absorbed mainly through lower margins rather than further price increases.

Notably, one might expect that a higher market wage level would increase household nominal purchasing power, allowing demand to be sustained at higher nominal prices. In the LBM, however, this effect is offset by the model's liquidity buffer and profit distribution mechanism. Since firms retain liquidity buffers as a fraction of their wage costs, higher levels of wages and employment can increase liquidity retained within firms and reduce the liquidity available to households. The fact that real consumption does not fall in the 1% adoption market, despite higher prices and potentially lower household nominal purchasing power, suggests that the presence of the learned firm improves the conversion of demand into realized sales. The lower mean unsatisfied demand ratio indicates that households are able to fulfill a larger share of their effective demand through their supplier connections. This improvement in demand fulfillment may help explain why real consumption remains slightly higher despite higher prices and greater liquidity retention.

Partial adoption coordinates the goods market better than full adoption The other puzzle in the extended LBM is the fact that the outcomes of the goods market do not improve monotonically with adoption, as shown in Section 4.3.2. One might expect that if adoption improves coordination from 0% to 1% and from 1% to 50%, then the 100% adoption market should be the most coordinated. However, the 100% adoption scenario is less coordinated than the 50% adoption scenario: real consumption is slightly lower, unsatisfied demand is higher, and the economy exhibits a persistent excess supply issue.

The model's liquidity buffer and profit distribution mechanism discussed above partly provide an explanation for this non-monotonicity. A higher-wage and higher-price regime creates downward pressure on real consumption, while higher employment increases production capacity. Because these two forces move in opposite directions, higher adoption does not necessarily translate into higher realized consumption. In the 1% and 50% adoption markets, improved goods-market coordination appears to dominate, as lower unsatisfied demand ratios coincide with more complete fulfillment of effective demand and slightly higher real consumption. However, as shown in

Table 14, unsatisfied demand worsens from 50% to 100% adoption, suggesting that the additional production capacity under full adoption is no longer translated into additional realized consumption.

A more precise question, then, is why unsatisfied demand does not decrease under 100% adoption despite much higher inventories. The first explanation is an aggregate capacity explanation. In the 0%, 1% and 50% adoption scenarios, the decline in the mean unsatisfied demand ratio coincides with a narrowing gap between aggregate production and real consumption. This pattern is consistent with additional production capacity helping households fulfill more of their effective demand. By the 50% adoption scenario, however, real consumption is already almost equal to aggregate production. This suggests that insufficient aggregate production is unlikely to be the primary remaining constraint by the 50% adoption scenario. Beyond this point, additional production is less likely to translate into additional real consumption, and is more likely to appear as inventory.

The second explanation is a local matching explanation. Even when the economy has enough goods in aggregate, those goods may not be available through the particular supplier connections used by households. The reported statistic is based on the effective demand that remains after households complete their daily purchasing attempts. It can therefore remain positive when firms outside the household's supplier set hold excess inventories, or when the household is unable to fulfill its remaining effective demand through its connected suppliers during the available purchasing attempts. This local mismatch can explain why the 100% adoption scenario exhibits both excess aggregate inventories and remaining unsatisfied demand.

5.3 Limitations and future directions

The main goal of the extended LBM is not to provide direct empirical predictions about real economies, but to study how reinforcement learning can be used to analyze firm behavior, systemic outcomes and internal mechanisms of a macroeconomic ABM. Therefore, the results and findings presented in Sections 4 and 5 should be interpreted with consideration of the limitations in the modeling choices, as well as training and evaluation methodologies.

Model scope and external validity The LBM is a deliberately stylized representation of a macroeconomy: it contains only one type of good and two kinds of agents, simplified behavioral rules, fixed production technology and a closed monetary

economy with fixed liquidity. While these choices make the model parsimonious and, therefore, useful for isolating the effects of firm behavior, they also limit the external validity of the results. In real-world markets, firms compete not only through prices and wages, but also through product differentiation, operating efficiency, institutional constraints and many other factors that contribute to profitability. Because these channels are unavailable in the LBM, the learned strategy of lower-price, higher-wage and scale-expansion should not be interpreted as a general prediction about how real firms would behave under profit optimization. Instead, it shows what kind of strategy can emerge under the specific modeling choices of the LBM, as well as the specific properties of the LBM that allow that strategy to be profitable.

The principles of reinforcement learning and its application to agent behavior, however, are general enough that the methodology employed in this thesis can be applied to more complex macroeconomic ABMs. Future work that utilizes models with richer behavioral and institutional dimensions may improve predictive capability and external validity, likely in exchange for lower tractability.

Training setup and reward specification Because the learned policy was trained under the single-learned-firm setup, the behavior of learned firms in the 50% and 100% adoption scenarios does not necessarily represent the most profitable policy available under those market conditions. While this design choice is useful for studying the effect of scaled deployment of the policy learned in the 1% adoption scenario, and to some extent, the evolution of the market as other firms adopt the leading strategy, it does not represent a multi-agent equilibrium where all learned firms are simultaneously optimized for profit seeking. Future work that trains firms using a multi-agent reinforcement learning framework could provide more insight into what kind of strategy would emerge when learned firms adapt to other learned firms, as well as the systemic effects of genuine multi-agent profit maximization.

Relatedly, the notion of profit maximization and the reward specification implemented in this thesis should not be considered the definitive interpretation of what a firm should achieve. There are components in the reward function that encourage sustainable long-run survival and prevent early collapse in the interest of making firm-level and system-level analysis more straightforward. Different reward functions that focus on different aspects of the profit formula, such as profit margin or distributed profit as a proxy for returns on investment, could result in different learned strategies and, consequently, different systemic outcomes.

Observation and evaluation In order to improve the comparability between the learned policy and the baseline heuristic, the observation space of the learned firm is deliberately limited to local information, excluding information about competitors or aggregate market statistics. While this design choice fits within the context of the LBM, it also limits the interpretation of the learned strategy. The analysis in this thesis infers the learned strategy from observed behavior and market statistics, which is a simplification of the internal and more complex decision process of the recurrent policy. Without further analysis into the inner workings of the learned policy, it is hard to pinpoint exactly how the learned firm utilizes the observed signals to make decisions. Future work could expand on this limitation by employing interpretability tools to gain a better understanding of the learned policy and strategy.

Additionally, the learned policy is trained and evaluated mainly under one set of model parameters. While this does not pose an issue for the main evaluation in this thesis, since the evaluation models use the same parameters, it limits the ability to determine whether the learned policy generalizes across different economic environments. It is also difficult to tell whether the policy learns a general firm strategy or exploits specific parameter values of the training environment. In addition to using interpretability tools to better understand the behavior of the learned firm, training the learned policy under models with varied parameter sets could improve the ability of the resulting policy to generalize across economies with different characteristics. More extensive robustness checks across seeds, parameter settings and evaluation horizons could further test the stability of the observed patterns.

Lastly, the system-level analysis of this thesis focuses on important macroeconomic variables such as unemployment, wages and unsatisfied demand, but these metrics do not amount to a complete welfare analysis. Although certain directions of change in the goods market and labor market can be identified, it is difficult to determine whether the economy as a whole improves or worsens. Future analysis could expand the macroeconomic results with explicit welfare metrics and household-level measures, such as liquidity inequality, consumption inequality or unemployment duration.

6 Conclusion

This thesis examined how reinforcement learning can be used to extend a macroeconomic agent-based model by training firm behavior with nominal profit as the primary objective. Specifically, it replaced one firm in the Lengnick baseline model with a learned firm controlled by a recurrent policy trained using Proximal Policy Optimization. The resulting policy and its effects on the macroeconomy were analyzed and interpreted to study both the firm-level and system-level outcomes of introducing profit-oriented learned firms into the baseline model. Methodologically, the results demonstrate that a recurrent PPO policy can control a firm's mixed continuous and discrete decisions within a macroeconomic ABM while operating under partial observability.

The results show that reinforcement learning can produce a firm policy that substantially outperforms the baseline heuristic in generating profits under the conditions of the Lengnick baseline model. Not only does the learned firm make more profit both monthly and cumulatively, but it also typically becomes much larger, longer-lived and more financially stable, capturing a disproportionate share of market activity relative to baseline firms. The analysis of the learned firm's behavior suggests that the superior policy can be understood as a lower-margin, scale-expansion strategy, rather than a simple high-price or low-wage approach. The learned firm tends to price below the market price while paying above the market wage, sacrificing profit margin for a much larger customer and employee base than a typical active firm. This outcome demonstrates the existence of a sustainable profit generation strategy within the context of the Lengnick baseline model, one that focuses on size and scale rather than short-term margin extraction.

The macroeconomic effects of introducing learned firms into the economy were analyzed across different adoption scenarios, corresponding to the share of the firm population that follows the learned policy. Higher adoption levels improve the labor market through lower and more stable unemployment, and are associated with lower levels of unsatisfied demand and a more coordinated goods market. However, the effects of the learned policy on the goods market are nonlinear: partial adoption coordinates the market better than full adoption, and the 100% adoption market is disrupted by a persistent excess supply issue. The market structure statistics also show that, while learned firms dominate baseline firms under partial adoption, higher adoption levels are associated with lower aggregate profitability and a weaker per-firm profit advantage for learned firms. Beyond documenting these outcomes, this thesis

highlights the benefits of the agent-based modeling framework for understanding emergent system-level patterns, which do not necessarily follow mechanically from agent-level behavior.

The findings should nevertheless be interpreted within the limitations of the model, training setup and evaluation protocol. The external validity of the results is limited by the simplicity and parsimony of the Lengnick baseline model. Because the 50% and 100% adoption scenarios represent scaled deployments of the learned policy trained in the 1% adoption scenario, rather than genuine multi-agent optimization, the behavior of learned firms in those scenarios does not necessarily represent the most profitable policy available under those market conditions. The interpretation of the learned strategy is also inferred mainly through observed patterns and outcomes, rather than through a dedicated interpretability analysis of the recurrent policy.

Future work could expand on this thesis by addressing the limitations above, including through a more behaviorally rich model, a multi-agent reinforcement learning framework or interpretability tools for the behavioral analysis. Additionally, the robustness and generalization of the policy could be improved by adding more variety to the training environment. Lastly, alternative reward specifications and a more comprehensive welfare analysis could provide a more complete picture of the firm-level and system-level outcomes. In summary, the findings show that reinforcement learning can identify effective alternative firm behaviors within a macroeconomic agent-based model, while also revealing that the systemic consequences of an individually successful strategy depend on how widely it is adopted.

References

[1] C. Berner et al. *Dota 2 with Large Scale Deep Reinforcement Learning*. 2019. arXiv: 1912.06680 [cs.LG].

[2] E. Bonabeau. "Agent-Based Modeling: Methods and Techniques for Simulating Human Systems". In: *Proceedings of the National Academy of Sciences* 99.Suppl. 3 (2002), pp. 7280–7287.

[3] S. Brusatin et al. "Simulating the Economic Impact of Rationality through Reinforcement Learning and Agent-Based Modelling". In: *Proceedings of the 5th ACM International Conference on AI in Finance*. Association for Computing Machinery, 2024, pp. 159–167. DOI: 10.1145/3677052.3698621.

[4] G. A. Calvo. "Staggered Prices in a Utility-Maximizing Framework". In: *Journal of Monetary Economics* 12.3 (1983), pp. 383–398.

[5] A. Charpentier, R. Élie, and C. Remlinger. "Reinforcement Learning in Economics and Finance". In: *Computational Economics* 62.1 (2023), pp. 425–462. DOI: 10.1007/s10614-021-10119-4.

[6] H. Dawid and D. Delli Gatti. "Agent-Based Macroeconomics". In: *Handbook of Computational Economics*. Ed. by C. Hommes and B. LeBaron. Vol. 4. Elsevier, 2018, pp. 63–156. DOI: 10.1016/bs.hescom.2018.02.006.

[7] K. Dwarakanath, T. Balch, and S. Vyetrenko. *ABIDES-Economist: Agent-Based Simulator of Economic Systems with Learning Agents*. 2024. DOI: 10.48550/arXiv.2402.09563. arXiv: 2402.09563 [cs.MA]. URL: https://arxiv.org/abs/2402.09563.

[8] M. Hausknecht and P. Stone. "Deep Recurrent Q-Learning for Partially Observable MDPs". In: *AAAI Fall Symposium Series*. 2015. arXiv: 1507.06527 [cs.LG].

[9] S. Hochreiter and J. Schmidhuber. "Long Short-Term Memory". In: *Neural Computation* 9.8 (1997), pp. 1735–1780. DOI: 10.1162/neco.1997.9.8.1735.

[10] E. ter Hoeven et al. "Mesa 3: Agent-Based Modeling with Python in 2025". In: *Journal of Open Source Software* 10.107 (2025), p. 7668. DOI: 10.21105/joss.07668.

[11] S. Huang et al. “CleanRL: High-Quality Single-File Implementations of Deep Reinforcement Learning Algorithms”. In: *Journal of Machine Learning Research* 23.274 (2022), pp. 1–18. URL: http://jmlr.org/papers/v23/21-1342.html.

[12] L. P. Kaelbling, M. L. Littman, and A. R. Cassandra. “Planning and Acting in Partially Observable Stochastic Domains”. In: *Artificial Intelligence* 101.1–2 (1998), pp. 99–134. DOI: 10.1016/S0004-3702(98)00023-X.

[13] T. Kluyver et al. “Jupyter Notebooks – a Publishing Format for Reproducible Computational Workflows”. In: *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. IOS Press, 2016, pp. 87–90. DOI: 10.3233/978-1-61499-649-1-87.

[14] J. Kober, J. A. Bagnell, and J. Peters. “Reinforcement Learning in Robotics: A Survey”. In: *The International Journal of Robotics Research* 32.11 (2013), pp. 1238–1274. DOI: 10.1177/0278364913495721.

[15] H. Le. *Reinforcement Learning for Firm Behavior in a Macroeconomic Agent-Based Model*. Version 1.0.0. 2026. DOI: 10.5281/zenodo.21674873. URL: https://doi.org/10.5281/zenodo.21674873.

[16] M. Lengnick. “Agent-Based Macroeconomics: A Baseline Model”. In: *Journal of Economic Behavior & Organization* 86 (2013), pp. 102–120.

[17] S. Levine et al. *Offline Reinforcement Learning: Tutorial, Review, and Perspectives on Open Problems*. 2020. arXiv: 2005.01643. URL: https://arxiv.org/abs/2005.01643.

[18] M. Mitchell. *Complexity: A Guided Tour*. Oxford University Press, 2009.

[19] T. M. Moerland et al. *Model-Based Reinforcement Learning: A Survey*. 2020. arXiv: 2006.16712 [cs.LG]. URL: https://arxiv.org/abs/2006.16712.

[20] OpenAI et al. “Learning Dexterous In-Hand Manipulation”. In: *The International Journal of Robotics Research* 39.1 (2020), pp. 3–20. DOI: 10.1177/0278364919887447.

[21] A. Paszke et al. “PyTorch: An Imperative Style, High-Performance Deep Learning Library”. In: *Advances in Neural Information Processing Systems*. Vol. 32. Curran Associates, Inc., 2019, pp. 8024–8035. URL: https://

papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.

[22] S. F. Railsback and V. Grimm. *Agent-Based and Individual-Based Modeling: A Practical Introduction*. Princeton University Press, 2019.

[23] J. Schulman et al. "High-Dimensional Continuous Control Using Generalized Advantage Estimation". In: *Proceedings of the International Conference on Learning Representations (ICLR)*. 2016.

[24] J. Schulman et al. *Proximal Policy Optimization Algorithms*. 2017. arXiv: 1707.06347 [cs.LG].

[25] R. S. Sutton and A. G. Barto. *Reinforcement Learning: An Introduction*. 2nd ed. The MIT Press, 2018. URL: http://incompleteideas.net/book/the-book-2nd.html.

[26] L. Tesfatsion. "Agent-Based Computational Economics: Growing Economies from the Bottom Up". In: *Artificial Life* 8.1 (2002), pp. 55–82. DOI: 10.1162/106454602753694765.

[27] M. Towers et al. *Gymnasium: A Standard Interface for Reinforcement Learning Environments*. 2024. arXiv: 2407.17032 [cs.LG].

[28] W. Zhang, A. Valencia, and N.-B. Chang. "Synergistic Integration Between Machine Learning and Agent-Based Modeling: A Multidisciplinary Review". In: *IEEE Transactions on Neural Networks and Learning Systems* 34.5 (2023), pp. 2170–2190. DOI: 10.1109/TNNLS.2021.3106777.

[29] S. Zheng et al. "The AI Economist: Taxation Policy Design via Two-Level Deep Multiagent Reinforcement Learning". In: *Science Advances* 8.18 (2022), eabk2607. DOI: 10.1126/sciadv.abk2607.