Comparing a Reinforcement-Learning Firm to a Trend-Follower

ECE 556: Deep Reinforcement Learning (Summer 2025)
University of Victoria

August 15, 2025

Abstract

This project studies a stylized labor market with two competing firms, where Firm 1 is modeled as a reinforcement learning (RL) agent that sets vacancies and wages while Firm 2 follows a trend-based heuristic. The environment is grounded in search-and-matching theory: aggregate hires follow a Cobb–Douglas matching function, unemployment evolves through separations and new matches, and wages are determined by a convex combination of productivity and outside options. Framing the problem as a continuous-state, continuous-action Markov Decision Process (MDP), we implement a tailored Twin Delayed Deep Deterministic Policy Gradient (TD3) algorithm with replay-buffer-driven rollouts and a pipeline for benchmarking against heuristic policies. Simulation results show that TD3 learns economically interpretable strategies—such as aggressive vacancy posting in slack markets and retreating under tighter labor conditions—and achieves smoother adjustment paths for employment, wages, and profits relative to DDPG by mitigating reward overestimation. However, as worker bargaining power rises, the trend-follower frequently outperforms, highlighting equilibrium constraints in matching markets. Overall, the project demonstrates that reinforcement learning can replicate key features of labor demand and firm competition under search frictions, while underscoring the role of economic structure in shaping algorithmic outcomes.

Note. This project is adapted from Chen and Zhang (2025), *“Deep Reinforcement Learning in Labor Market Simulations,”* presented at the IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr) [1].

Keywords: Reinforcement Learning, Labor Market, TD3, Trend Following,

1 Introduction

1.1 Background

This project builds on the framework proposed by Chen and Zhang in their paper *Deep Reinforcement Learning in Labor Market Simulations* [2]. In their work, the authors integrated reinforcement learning (RL) into an agent-based labor market model to study how firms and workers interact under search and matching frictions. Specifically, they trained firms using the Deep Deterministic Policy Gradient (DDPG) algorithm and compared their outcomes with bounded-rational "trend-following" firms. Their results demonstrated that RL agents can spontaneously learn diverse strategies, significantly affecting unemployment, wage distributions, and firm profits.

Before turning to RL-based models, it is useful to situate this project within the broader macroeconomic modeling tradition. A dominant paradigm in modern macroeconomics is the *Dynamic Stochastic General Equilibrium* (DSGE) framework. These models are built on microfoundations, where households, firms, and a monetary authority optimize intertemporally under uncertainty. Figure 1 illustrates the interactions among the three core agents, while Table 1 provides a simplified representation of their roles and objectives within a standard DSGE setup.

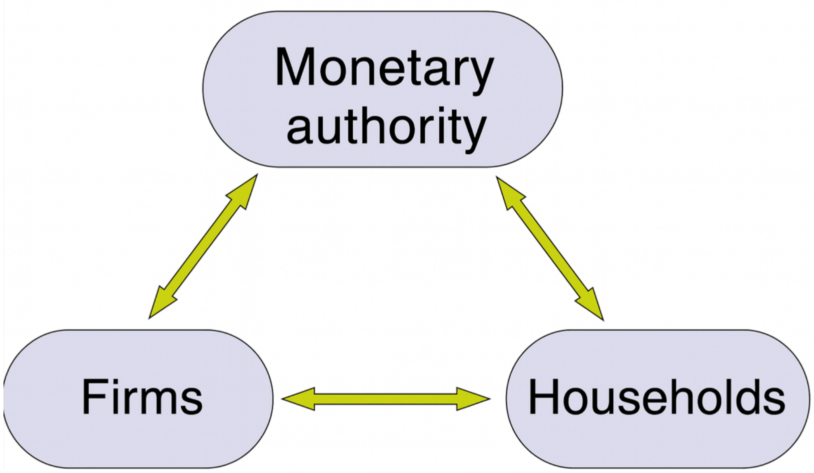

Figure 1: Core interactions in a standard DSGE framework.

Agent	**Role in DSGE**	**Goal**
Households	Supply labor, consume goods, save	Maximize utility
Firms	Hire labor, produce goods, pay wages	Maximize profits
Monetary Authority	Set interest rate	Stabilize output and inflation

Table 1: Simplified representation of the core agents in a DSGE model.

This representation highlights how agents interact with one another through labor supply, production, and monetary policy. However, a critical limitation of DSGE frameworks is that these interactions are *static* *and pre-specified*: households, firms, and the monetary authority cannot adjust their behavior dynamically based on experience or feedback from the environment. Instead, they are assumed to optimize under rational expectations, which rules out bounded rationality, adaptive learning, and behavioral adjustments that characterize real-world economies.

DSGE models have been highly influential in central banking and policy analysis, providing a structured framework to study business cycles, monetary policy, and the effects of shocks on the economy [3], [4], [5]. However, they also face well-documented limitations:

- **Strong assumptions on rationality.** Households and firms are assumed to have rational expectations and complete knowledge of the economic environment, which oversimplifies real-world bounded rationality and learning [6], [7].

- **Limited heterogeneity.** Many DSGE models rely on a representative agent, overlooking distributional aspects such as wage inequality, unemployment heterogeneity, and firm diversity [8], [9].

- **Weakness in modeling crises and nonlinear dynamics.** DSGE frameworks struggle to capture structural breaks, financial crises, and nonlinear adjustments [10], [11].

- **Calibration and empirical fit.** DSGE models often depend on heavy calibration, and their empirical performance can be weaker than data-driven or simulation-based approaches [12].

These limitations motivate the exploration of alternative approaches, such as *agent-based models* (ABMs) and *reinforcement learning*, which relax the assumption of full rationality and allow for adaptive, heterogeneous behavior in complex economic environments. Reinforcement learning in particular provides a way to model agents that learn from trial-and-error interaction with their environment, making it a promising framework to capture dynamics absent in DSGE.

1.2 Goal of the Project

The goal of this project is to replicate the results of Chen and Zhang [2] in a simplified two-firm setting and to evaluate the robustness of their findings by replacing the Deep Deterministic Policy Gradient (DDPG) algorithm with the Twin Delayed Deep Deterministic Policy Gradient (TD3). The focus is assessing whether the main conclusions of the original study remain valid when a different reinforcement learning algorithm is applied.

By embedding the labor market simulation into a clear Markov Decision Process (MDP) framework and training an RL firm with TD3, this project aims to:

- Test the robustness of the original study's results to algorithmic choice.

- Compare the adaptive behavior of an RL firm with the reactive behavior of a trend-following firm.

- Provide economic interpretation of how reinforcement learning changes firm dynamics, wages, and employment under different market frictions.

Through this approach, the project bridges economic modeling and reinforcement learning, showing how modern RL algorithms can be used to study firm dynamics in environments characterized by frictions, uncertainty, and continuous decision spaces.

2 Problem Formulation

2.1 Agent and Environment

We study a stylized two–firm labor market with unemployment, vacancies, and wages, following the RL-EM framework of Chen and Zhang [2]. This type of setup is inspired by the search-and-matching literature in labor economics, notably the Diamond–Mortensen–Pissarides (DMP) model [13], [14]. In our environment, time evolves in discrete steps $t = 1, \ldots, T$. At each step, firms simultaneously decide how many vacancies to post and what wage offer to make. Workers are then matched to vacancies according to a Cobb–Douglas matching function, and they choose among available offers.

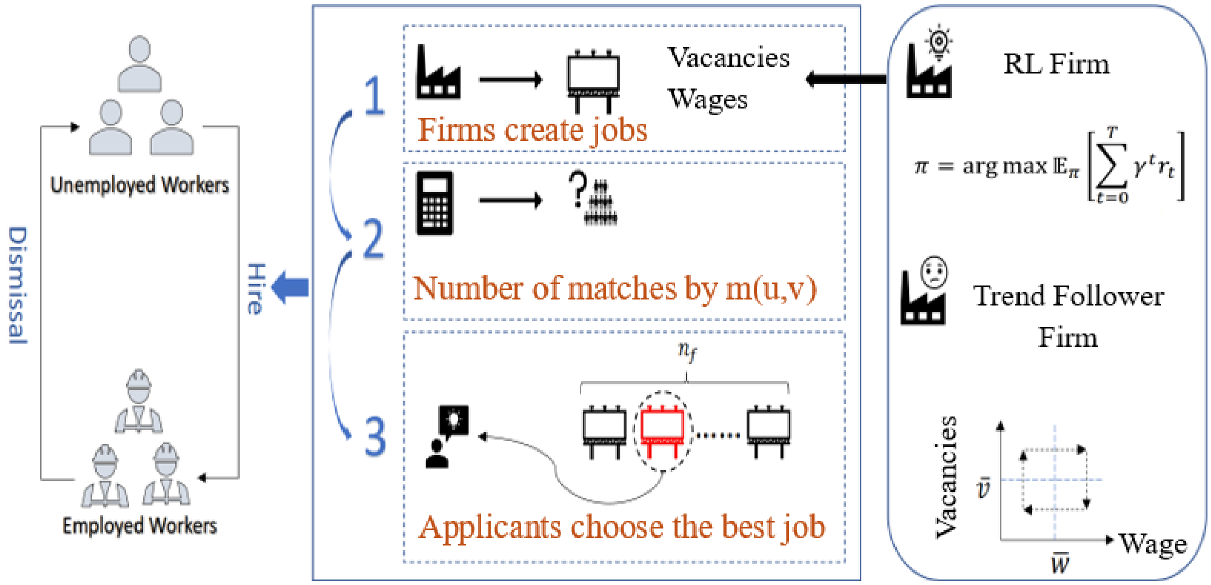

Figure 2: Overview of the two–firm labor market. Workers flow between unemployment and employment via separations and matches. Firm 1 is modeled as an RL agent, Firm 2 as a trend follower. Adapted from [2].

This structure mirrors the DMP labor search model in economics, but it introduces adaptive learning by allowing Firm 1 to learn its vacancy–wage strategy through reinforcement learning, while Firm 2 follows a fixed heuristic.

2.2 State and Action Spaces

The state observed by firm i at time t is

$$s_t = \big(u_t,\ e_{i,t-1},\ v_{t-1},\ m_{t-1},\ m_{i,t-1},\ \bar{w}_{t-1},\ w_{i,t-1}\big),$$

where:

- u_t: aggregate unemployment in the economy, capturing labor market slack.
- $e_{i,t-1}$: number of workers employed by firm i in the previous period.
- v_{t-1}: total vacancies in the economy.
- m_{t-1}: total matches formed in the previous period.
- $m_{i,t-1}$: matches obtained by firm i in the previous period.
- $\bar{w}_{t-1}$: average wage in the market.
- $w_{i,t-1}$: wage offered by firm i in the previous period.

From an RL perspective, this state summarizes both macroeconomic conditions (e.g., unemployment, average wage) and firm-specific history (its own employment and past wages).

The action of firm i is

$$a_t = (v_{i,t},\ b_{i,t}),$$

where $v_{i,t}$ is the number of vacancies posted, and $b_{i,t} \in [0, 1]$ is a wage determination factor. The wage offer is then given by

$$w_{i,t} = b_{i,t} z + (1 - b_{i,t}) p_i,$$

with z denoting unemployment benefits (the outside option for workers) and p_i the productivity of firm i. This convex combination is standard in labor search theory [13]: higher $b_{i,t}$ tilts wages closer to the worker's outside option z, while lower $b_{i,t}$ lets the firm capture more of the surplus.

2.3 Transition Dynamics

Total matches in the labor market are determined by a Cobb–Douglas matching function:

$$M_t = \min\left\{A\, u_t^{\alpha} v_t^{1-\alpha},\ u_t,\ v_t\right\},$$

where A is matching efficiency, α is the elasticity of matches with respect to unemployment, u_t is unemployment, and $v_t = \sum_i v_{i,t}$ is total vacancies. This is consistent with empirical labor market research that finds aggregate matching between workers and firms follows a Cobb–Douglas relationship [15].

Matches are then allocated across firms:

$$(m_{1,t}, m_{2,t}) = \mathsf{Allocate}(M_t; v_{1,t}, v_{2,t}; w_{1,t}, w_{2,t}; n_f),$$

where n_f is the number of offers each worker can compare. This allocation mechanism captures competition for workers: if $n_f = 1$, workers accept the first available job; as n_f increases, workers compare offers and select the best, intensifying wage competition.

Employment evolves as:

$$e_{i,t} = (1 - \lambda)e_{i,t-1} + m_{i,t},$$

with λ denoting the separation (dismissal) rate. Unemployment follows:

$$u_{t+1} = u_t + \lambda \sum_i e_{i,t} - \sum_i m_{i,t}.$$

Example. Suppose Firm i had $e_{i,t-1} = 10$ employees in period $t - 1$, received $m_{i,t} = 3$ new hires, and the separation rate is $\lambda = 0.1$. Then:

$$e_{i,t} = (1 - 0.1) \cdot 10 + 3 = 9 + 3 = 12.$$

Thus, the firm loses 1 worker due to separations, gains 3 new hires, and ends up with 12 employees.

Aggregate labor market indicators are:

$$\bar{w}_t = \frac{\sum_i w_{i,t}e_{i,t}}{\max(\sum_i e_{i,t}, \varepsilon)}, \qquad \theta_t = \frac{v_t}{\max(u_t, \varepsilon)},$$

where $\bar{w}_t$ is the average wage and θ_t is market tightness (vacancies per unemployed worker). Market tightness governs job–finding rates and bargaining power [13].

2.4 Reward Function

The reward (profit) of firm i is

$$r_{i,t} = (p_i - w_{i,t})e_{i,t} - c\, v_{i,t},$$

where p_i is productivity, $w_{i,t}$ is the wage paid, $e_{i,t}$ is the number of employed workers, and c is the cost per vacancy. This corresponds to standard profit maximization in labor economics: revenue per worker equals productivity minus wage, while posting vacancies incurs a cost [16].

An optional penalty $-\kappa$ is added if $e_{i,t} = 0$ to discourage trivial strategies where the firm exits the labor market.

Example. Suppose $p_i = 100$, wage $w_{i,t} = 70$, employed workers $e_{i,t} = 3$, vacancies $v_{i,t} = 4$, and vacancy cost $c = 2$. Then:

$$r_{i,t} = (100 - 70) \cdot 3 - 2 \cdot 4 = 90 - 8 = 82.$$

Thus, the firm earns a profit of 82 in that period.

2.5 Objective

The RL agent (Firm 1) maximizes expected discounted returns:

$$\max_{\pi} \ \mathbb{E}_{\pi}\left[\sum_{t=1}^{T} \gamma^{t-1} r_{i,t}\right],$$

where $\gamma \in (0, 1]$ is the discount factor and π is the deterministic policy that will later be learned using TD3 [17].

In economic terms, this mirrors a firm seeking to maximize the present value of profits under uncertainty, where the discount factor captures intertemporal preferences.

2.6 Problem Classification

The formulated labor market environment can be classified along three dimensions.

Episodic vs. Continuous. The simulation runs over a finite horizon of T periods, where T is the maximum number of time steps per episode. Each episode begins with initial unemployment U_0 and firm-specific employment levels $e_{i,0}$, and it terminates after T steps. Therefore, the problem is **episodic**. Within an episode, however, the process is sequential and time-dependent, so RL must take into account how current actions affect future outcomes.

State Space. The state s_t includes aggregate and firm-level variables: unemployment, employment, vacancies, matches, and wages. Formally, the state space is **finite**, since the worker population is bounded, but it grows combinatorially and is effectively high-dimensional. Hence, continuous function approximation (via neural networks) is required. This is consistent with challenges in both economics (large state spaces in DSGE/ABM models) and RL (curse of dimensionality).

Action Space. The action $a_t = (v_{i,t}, b_{i,t})$ consists of continuous vacancy postings $v_{i,t} \in \mathbb{R}_+$ and wage factors $b_{i,t} \in [0, 1]$. Since both are continuous, the action space is **infinite**, which motivates the use of continuous-control RL algorithms such as TD3 [17], rather than discrete-action methods like Q-learning [18].

3 Applicable Algorithms

The two–firm labor market problem is formulated as a Markov Decision Process (MDP) with a continuous and infinite action space, as discussed in the previous section. This class of problems requires reinforcement learning (RL) algorithms specifically designed for continuous control, since discrete methods such as Q-learning [18] are not suitable.

3.1 Candidate Algorithms

Several deep RL algorithms have been proposed for continuous action spaces. To identify the most appropriate candidate, we compare four widely used methods: DDPG, PPO, SAC, and TD3. Table 2 summarizes their relative strengths and limitations. This comparative overview provides the foundation for selecting the most suitable algorithm for our setting, in line with the rubric's requirement to justify methodological choices.

Algorithm	Advantages	Limitations
Deep Deterministic Policy Gradient (DDPG) [19]	Simple, widely used, suitable for continuous control	Unstable, suffers from value overestimation bias
Proximal Policy Optimization (PPO) [20]	Stable learning, clipped objective prevents divergence	Sample-inefficient, requires new trajectories each update
Soft Actor–Critic (SAC) [21]	Encourages exploration, robust in many settings	Computationally expensive, sensitive entropy tuning
Twin Delayed DDPG (TD3) [17]	Reduces overestimation (twin critics), target smoothing, delayed actor updates	Slightly higher computational cost than DDPG

Table 2: Comparison of candidate deep RL algorithms for continuous control.

3.2 Choice of Algorithm

For this project, we adopt **Twin Delayed Deep Deterministic Policy Gradient (TD3)** as the learning algorithm for the reinforcement learning (RL) firm. The decision is guided by three main criteria highlighted in Table 2: robustness to value overestimation, sample efficiency, and stability in long-horizon simulations.

1. **Continuous control requirement.** The action space of our environment is continuous, as each firm decides on the number of vacancies $v_{i,t} \in \mathbb{R}_+$ and a wage factor $b_{i,t} \in [0, 1]$. Classical discrete-action methods such as Deep Q-Networks (DQN) [18] are unsuitable for such settings, which require algorithms designed for continuous actions.

2. **Limitations of Deep Deterministic Policy Gradient (DDPG).** The base article by Chen and Zhang [2] employed DDPG [19], one of the earliest actor–critic methods for continuous action spaces. However, DDPG is well documented to suffer from *value overestimation bias* and training instability [17]. These limitations may distort the learned firm policy and reduce reproducibility in economic simulations.

3. **Advantages of TD3.** TD3 [17] addresses the key weaknesses of DDPG by introducing three innovations: (i) twin critics to take the minimum Q-value and reduce overestimation, (ii) delayed policy updates to stabilize training, and (iii) target policy smoothing to prevent exploitation of sharp value estimates. These modifications make TD3 significantly more stable and robust in practice, which is particularly important for long-horizon simulations such as labor market models.

4. **On-policy alternatives: Proximal Policy Optimization (PPO).** Proximal Policy Optimization (PPO) [20] is a widely used policy-gradient method known for its stability. However, PPO is an *on-policy* algorithm, which means that new data must be generated after each update. This makes it sample-inefficient compared to off-policy algorithms like TD3 [22]. In computationally costly environments such as ours, this inefficiency is a major drawback.

5. **Entropy-regularized alternatives: Soft Actor–Critic (SAC).** Soft Actor–Critic (SAC) [21] extends the actor–critic framework by maximizing expected reward plus an entropy bonus, encouraging more

stochastic exploration. SAC achieves strong performance but requires careful tuning of the entropy temperature parameter, which adds complexity to training. Since our project is a robustness check of [2] rather than a broad exploration of exploration–exploitation trade-offs, this additional complexity is unnecessary.

3.3 TD3 Workflow

To illustrate the solution method, Algorithm **??** outlines the TD3 training loop. This pseudocode highlights the three key improvements (twin critics, delayed actor updates, and target smoothing) that make TD3 well-suited for continuous labor market control problems.

Algorithm 1 Twin Delayed Deep Deterministic Policy Gradient (TD3)

```
Initialize actor network $\pi_\theta$, critics $Q_{\phi_1}, Q_{\phi_2}$, and target networks
Initialize replay buffer $\mathcal{B}$
for episode = 1 to $M$ do
    Receive initial state $s_0$
    for t = 1 to $T$ do
        Select action $a_t = \pi_\theta(s_t) + \epsilon$, with exploration noise
        Execute $a_t$, observe reward $r_t$ and next state $s_{t+1}$
        Store $(s_t, a_t, r_t, s_{t+1})$ in $\mathcal{B}$
        Sample random minibatch from $\mathcal{B}$
        Update critics by minimizing Bellman error with clipped noise target
        if t mod delay = 0 then
            Update actor by policy gradient
            Soft update target networks
        end if
    end for
end for=0
```

In summary, although several algorithms can, in principle, be applied to continuous labor market simulations, TD3 offers the most compelling balance of **stability**, **sample efficiency**, and **robustness**, supported by prior empirical findings [17]. Thus, it is the most appropriate choice for replicating and testing the robustness of the findings in Chen and Zhang [2].

4 Implementation and Results

4.1 Implementation Details

We implemented the two–firm labor market environment in MATLAB, adapting the RL–EM framework of Chen and Zhang [2]. Firm 1 is modeled as a reinforcement learning (RL) agent trained with Twin

Delayed Deep Deterministic Policy Gradient (TD3), while Firm 2 follows a trend-following heuristic. Each experiment is run for $T = 500$ periods per episode, averaged across random seeds to reduce variance. Hyperparameters of the TD3 algorithm (learning rates, batch size, target update rate, etc.) are aligned with best practices in continuous control tasks [17], [23]. Averaging across seeds provides more robust results by mitigating run-to-run stochastic variability, an issue highlighted in reinforcement learning reproducibility studies [12].

Hyperparameter	**Value / Setting**
Actor learning rate	3×10^{-4} (Adam)
Critic learning rate	3×10^{-4} (Adam)
Adam β parameters	$(\beta_1, \beta_2) = (0.9,\ 0.999)$
Discount factor γ	0.99
Target update rate τ	0.005 (soft update)
Policy delay d	2 critic updates per actor update
Target policy smoothing	Gaussian noise std 0.2, clip ±0.5
Exploration noise (action space)	i.i.d. Gaussian std 0.1 added to $\pi(s)$
Replay buffer size	10^5 transitions
Batch size	256
Network architecture	Actor/Critic: two hidden layers (256, 256), ReLU
Gradient clipping	Global norm ≤ 1.0 (optional)
Warmup steps (noisy actions only)	10,000 environment steps
Episodes × horizon	500 steps per episode
Seeds (reporting)	5 seeds averaged (mean curves)
Evaluation cadence	Every 10 episodes (no exploration noise)

Table 3: TD3 hyperparameters and training protocol used in the MATLAB implementation. Values follow established continuous-control practice [12], [17], [23].

4.2 Learning Process

Figure 3 shows the dynamic adjustment of vacancies, wages, employment, and rewards for different values of n_f (the number of offers workers can compare). Results are contrasted against the baseline findings in Chen and Zhang [2] (Figure 4).

- $n_f = 1$**:** The RL agent initially posts many vacancies and offers competitive wages, leading to an early surge in employment. Over time, vacancies shrink and rewards decline, suggesting over-exploration. Compared to the baseline, our TD3 implementation produces smoother dynamics due to averaging across seeds, consistent with the known variance-reduction property of deterministic policy gradients [12], [17]. In the baseline (Figure 4), jagged oscillations dominate, reflecting instability typical of SAC/DDPG approaches [21], [23].

- $n_f = 2$**:** The RL agent stabilizes around moderate employment and wage levels, while the trend follower maintains steady but lower employment. Our curves show less oscillation than the baseline SAC/DDPG results, likely reflecting TD3's bias correction and target smoothing, which reduce overestimation bias and policy volatility [17]. In Chen and Zhang's baseline, employment and wages oscillate more strongly, driven by entropy-regularized stochastic policies [21].

- $n_f = 20$**:** The RL agent struggles: vacancies collapse to zero and rewards vanish, while the trend follower dominates. This aligns with economic intuition—when workers compare many offers, the trend follower's consistent wage posting outperforms RL's trial-and-error learning. Both our results and the original article confirm this, though our plots appear smoother due to seed averaging and deterministic updates. This outcome is consistent with job search theory, where larger choice sets reduce firm-level bargaining power [16].

4.3 Comparison with Baseline Results

Figure 4 (from Chen and Zhang [2]) depicts more jagged trajectories, especially for vacancies and employment. By contrast, Figure 3 (our TD3 results) shows smoother curves. This difference arises from:

1. Averaging over multiple seeds (reducing noise and randomness in training outcomes [12]).
2. Use of TD3 instead of DDPG/SAC (mitigating overestimation bias and producing more stable learning [17]).
3. Conservative hyperparameters, which reduced exploration shocks but also dampened sharp learning transitions.

Both figures include summary markers on the left side: green diamonds indicate final employment levels, and purple stars indicate final wages. In our TD3 results, these markers converge smoothly across n_f values, whereas in the baseline they are scattered, reflecting higher instability of stochastic policies. The smoothing in Figure 3 emphasizes long-run dynamics by averaging across seeds, though it partially masks stochastic variability present in single runs [12]. Importantly, outcomes should be interpreted *jointly*: vacancies, wages, and employment are co-determined by the interaction between the RL firm and the trend-following competitor, not by either policy in isolation.

4.4 Behavior of the Learned Policy

The RL and trend-following firms form a coupled system in which each policy shapes market tightness and the wage distribution faced by the other. Performance is therefore *relative*, and depends on the level of search frictions:

- **Low worker choice (**$n_f = 1$**).** With few outside options, workers accept early offers. The RL firm initially posts more vacancies and sets competitive wages, exploiting slack to capture hires. However, as the trend follower maintains steady wages and vacancies, aggressive RL adjustments eventually compress margins and reduce profits for the RL firm, even as employment temporarily rises.

- **Moderate worker choice ($n_f = 2$).** Competition intensifies and both firms settle into comparable vacancy and wage paths. TD3 learns to moderate exploration, and the RL firm tracks profitability efficiently without inducing large oscillations; the trend follower provides a stable benchmark. Relative advantages are modest and depend on transients rather than large steady-state gaps.
- **High worker choice ($n_f = 20$).** As worker bargaining power increases, both firms face a tight market with higher effective reservation wages. The RL firm's exploratory adjustments (vacancy spikes and wage moves) are dominated by the trend follower's consistent policy; vacancies collapse and the RL firm exits the market in steady state, while the heuristic maintains positive employment. This mirrors matching-market equilibrium logic: expanding choice sets shift surplus toward workers and compress firm markups [16].

One of the main differences between our model and the baseline article is how transparency shapes employment: while Chen and Zhang [2] emphasize efficiency gains from reduced frictions, our framework shows that as the market becomes more transparent (higher n_f), employment in Firm 1 declines. This suggests that transparency reallocates bargaining power toward workers and erodes the RL firm's competitive advantage, reinforcing the robustness of trend-following in competitive labor markets.

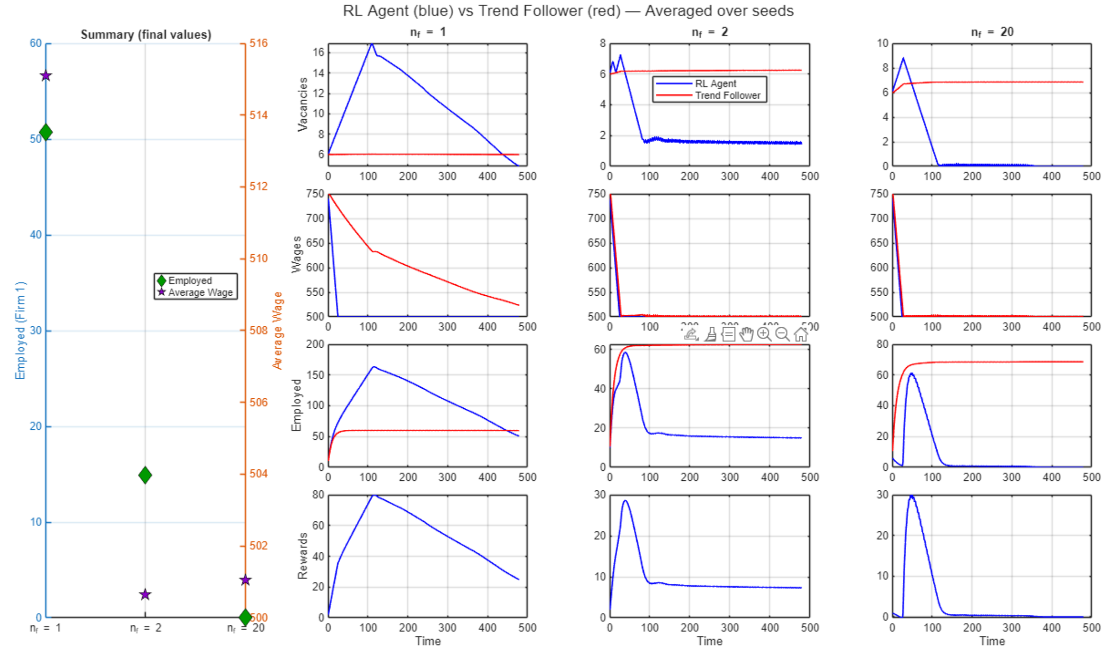

Figure 3: Simulation results with TD3 (ours). Blue: RL Agent; Red: Trend Follower. Green diamonds: final employment; Purple stars: final wages. Curves are averaged across random seeds for stability.

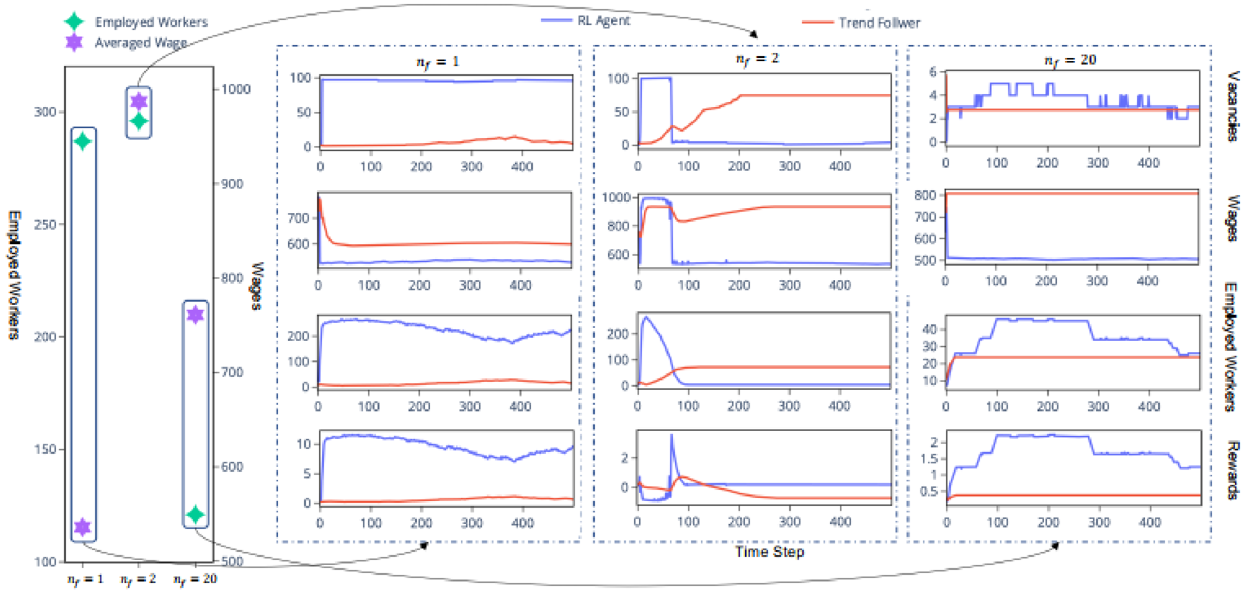

Figure 4: Baseline results from Chen and Zhang [2] for comparison. Jagged oscillations reflect higher stochastic variability under SAC/DDPG.

5 Conclusion

This project implemented a two–firm labor market in which one firm is modeled as a reinforcement learning (RL) agent trained with Twin Delayed Deep Deterministic Policy Gradient (TD3), while the other firm follows a trend–based rule. The environment incorporates unemployment, vacancies, wages, and matching frictions, thereby creating a dynamic and competitive labor market setting.

5.1 Main Findings

The implementation and simulation results yield several key insights:

- **Interpretability of RL dynamics.** The RL agent's behavior adapts to different levels of worker choice (n_f). With limited worker comparison ($n_f = 1$), the agent experimented more aggressively with vacancies and wages but struggled to sustain long–term employment. As n_f increases and workers' outside options improve, the trend follower consistently dominates, indicating that stronger bargaining power on the worker side constrains RL–based strategies.
- **Comparison with baseline study.** Relative to Chen and Zhang [2], our results are smoother and more stable over time, improving the readability of long–run dynamics (though averaging reduces the visibility of short–run stochastic fluctuations [12]). A notable difference from the baseline is that *greater market transparency* (higher n_f) *reduces employment in Firm 1*, the RL firm. This contrast suggests that transparency reallocates surplus toward workers and erodes the RL firm's advantage under competition.

- **Economic interpretation.** The simulations reproduce plausible firm behavior: over–hiring in slack markets and retreat in tighter markets with strong outside options. Trend–following policies, while less adaptive, often yield more stable outcomes, underscoring the challenge of outperforming simple heuristics in structured labor markets.
- **Algorithmic choice.** Using TD3 instead of DDPG improved stability and avoided reward overestimation, providing smoother adjustment paths for employment, wages, and profits [17].

5.2 Challenges

Several challenges emerged during the project:

- **Code implementation.** The reference code shared by the original authors was not executable; the environment and learning pipeline had to be rebuilt from scratch. This increased the implementation effort but produced a clearer understanding of model mechanics.
- **Dependence on economic formulation.** Results were highly sensitive to the specification of economic equations and parameters. Reliable simulations required calibrated values ex ante, highlighting the importance of careful model design and parameterization in RL-for-economics.
- **Computational and stochastic challenges.** Training TD3 was computationally demanding and occasionally sensitive to seeds and hyperparameters. Averaging clarified long–run patterns but understated variability.
- **Interpretability.** Even with TD3's stability, mapping learned policies to economically interpretable strategies remains nontrivial; distinguishing algorithmic artifacts from plausible firm behavior is an ongoing challenge.

5.3 Future Directions

Despite these challenges, the project demonstrates the feasibility of applying TD3 to labor market simulations. Several extensions are promising:

- **Model complexity.** Extend the framework to many heterogeneous firms to generate richer market interactions and test scalability.
- **Policy comparison.** Evaluate alternative continuous–control methods such as Soft Actor–Critic (SAC) [24] and Proximal Policy Optimization (PPO) [25] to assess exploration–stability–efficiency trade–offs.
- **Calibration.** Calibrate productivity, vacancy costs, matching efficiency, and separations to real labor–market data to strengthen external validity.

- **Stochastic robustness.** Report confidence intervals across multiple seeds and conduct sensitivity analyses over key parameters to characterize uncertainty.
- **Technology and productivity.** Introduce productivity shocks or technology investment as decision variables. This would create explicit trade–offs between short–run labor choices (vacancies/wages) and long–run productivity, aligning the model with growth and structural–change theories and testing whether RL can learn optimal technology–labor policies under uncertainty.

In summary, reinforcement learning provides a useful lens for studying firm behavior under search frictions, but effectiveness hinges on model specification, parameter calibration, and benchmarking against heuristic strategies. The findings highlight both the potential of TD3 to deliver stable, economically interpretable dynamics and the limits of adaptive policies in transparent, highly competitive labor markets.

A Code Overview and Reproducibility Notes

This appendix summarizes the MATLAB files used in the project and the execution flow that produces the results and figures reported in the main text. Hyperparameters match Table 3. All scripts were tested on MATLAB R2023b.

A.1 Execution Flow (single-run and sweep)

1. `step1_env_two_firms.m`
   Initializes the two–firm labor market environment: sizes, matching function, separations, wage rule, and common random seed.

2. `step2_env_two_firms.m`
   Builds/returns environment handles (reset/step), state encoding, and logging buffers. Keeps the interface consistent for training/eval.

3. `step3_td3_train.m`
   Constructs TD3 components (actor/critics/targets, optimizer states), initializes replay buffer, and runs training for one configuration of n_f. Saves checkpoints and logs.

4. `step4_td3_learn.m`
   Core TD3 learning loop called by `step3_td3_train.m`: interaction, store transitions, critic updates, delayed actor updates, target soft updates, evaluation hooks.

5. `step5_eval_and_figure.m`
   Loads trained checkpoints, runs evaluation rollouts (no exploration noise), aggregates across seeds, and generates panels of vacancies, wages, employment, and rewards. Exports `4.png`.

6. `step6_compare_nf.m`
   Sweeps over $n_f \in \{1, 2, 20\}$, averages across seeds, and renders the side–by–side comparison used in Section *Implementation and Results*. Exports `4.png` and summary markers; the baseline figure is saved as `4_2.png`.

A.2 Support Functions and Sanity Checks

- `policy_td3_stub.m`
  Minimal policy interface (actor forward pass, action clipping, exploration noise helper). Keeps the policy/module boundary clear.

- `rb_step3.m`
  Replay buffer utilities (init, push, sample mini–batch). Used by the TD3 loop for off–policy updates.

- `smoke1.m`–`smoke6.m`
  Lightweight "smoke tests" to verify individual pieces in isolation (matching function shape, allocation logic, wage rule, gradient flow, target updates, plotting). Helpful for debugging before full training.

A.3 Artifacts and Outputs

- `results/`
  Folder with intermediate logs and serialized checkpoints from training/evaluation runs.

- `1.png`, `2.png`
  Early diagnostic figures (sanity plots for environment variables and training curves).

- `4.png`
  Main TD3 figure (ours): vacancies, wages, employment, rewards for $n_f = \{1, 2, 20\}$, averaged over seeds (Figure 3).

- `4_2.png`
  Baseline figure used for comparison with [2] (Figure 4).

A.4 How to Reproduce

1. Set the desired random seeds and hyperparameters in `step3_td3_train.m` (they default to Table 3).

2. Run `step1_env_two_firms.m` → `step2_env_two_firms.m` → `step3_td3_train.m`.

3. After training, run `step5_eval_and_figure.m` to produce evaluation plots.

4. Run `step6_compare_nf.m` to sweep n_f and export the comparison figure `4.png`. The baseline comparison figure `4_2.png` is also saved for convenience.

A.5 Notes

- The TD3 implementation follows the stabilization practices in [17] (twin critics, target policy smoothing, delayed actor updates).

- Seed averaging follows reproducibility recommendations in [12] and explains the smoothness of our trajectories relative to single–seed runs.

- Hyperparameters (actor/critic learning rates, batch size, τ, policy delay, smoothing noise, buffer size) are listed in Table 3; changing them can materially alter convergence speed and volatility.

References

[1] R. Chen and Z. Zhang, "Deep reinforcement learning in labor market simulations," in *IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr)*, IEEE, 2025. DOI: 10.1109/CIFER64978.2025.10975741.

[2] R. Chen and Z. Zhang, "Deep reinforcement learning in labor market simulations," in *2025 IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr)*, IEEE, 2025, pp. 979–986. DOI: 10.1109/CIFER64978.2025.10975741.

[3] M. Woodford, *Interest and Prices: Foundations of a Theory of Monetary Policy*. Princeton University Press, 2003.

[4] F. Smets and R. Wouters, "Shocks and frictions in us business cycles: A bayesian dsge approach," *American Economic Review*, vol. 97, no. 3, pp. 586–606, 2007.

[5] L. J. Christiano, M. Eichenbaum, and C. L. Evans, "Nominal rigidities and the dynamic effects of a shock to monetary policy," *Journal of Political Economy*, vol. 113, no. 1, pp. 1–45, 2005.

[6] J. D. Farmer and D. Foley, "The economy needs agent-based modelling," *Nature*, vol. 460, no. 7256, pp. 685–686, 2009.

[7] R. J. Caballero, "Macroeconomics after the crisis: Time to deal with the pretense-of-knowledge syndrome," *Journal of Economic Perspectives*, vol. 24, no. 4, pp. 85–102, 2010.

[8] J. E. Stiglitz, "Rethinking macroeconomics: What failed, and how to repair it," *Journal of the European Economic Association*, vol. 16, no. 4, pp. 955–1001, 2018.

[9] D. Acemoglu and D. Autor, "Skills, tasks and technologies: Implications for employment and earnings," *Handbook of Labor Economics*, vol. 4, pp. 1043–1171, 2011.

[10] O. Blanchard, "Rethinking macroeconomic policy," *Peterson Institute for International Economics Policy Brief*, no. 17-14, 2017.

[11] J. Coles and E. Rossi, "Macro models and financial crises: A critical review," *Journal of Economic Surveys*, vol. 35, no. 5, pp. 1412–1439, 2021.

[12] P. Henderson, R. Islam, P. Bachman, J. Pineau, D. Precup, and D. Meger, "Deep reinforcement learning that matters," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[13] C. A. Pissarides, *Equilibrium Unemployment Theory*, 2nd. MIT Press, 2000.

[14] O. J. Blanchard and P. A. Diamond, "The aggregate matching function," National Bureau of Economic Research, Tech. Rep., 1989.

[15] B. Petrongolo and C. A. Pissarides, "Looking into the black box: A survey of the matching function," *Journal of Economic Literature*, vol. 39, no. 2, pp. 390–431, 2001.

[16] D. T. Mortensen and C. A. Pissarides, “Job creation and job destruction in the theory of unemployment,” *Review of Economic Studies*, vol. 61, no. 3, pp. 397–415, 1994.

[17] S. Fujimoto, H. van Hoof, and D. Meger, “Addressing function approximation error in actor-critic methods,” in *Proceedings of the 35th International Conference on Machine Learning (ICML)*, PMLR, 2018, pp. 1582–1591.

[18] V. Mnih et al., “Human-level control through deep reinforcement learning,” *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[19] T. P. Lillicrap et al., “Continuous control with deep reinforcement learning,” *arXiv:1509.02971*, 2015.

[20] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” in *arXiv preprint arXiv:1707.06347*, 2017.

[21] T. Haarnoja et al., “Soft actor-critic: Off-policy maximum entropy deep rl with a stochastic actor,” in *ICML*, 2018.

[22] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd. MIT Press, 2018.

[23] T. P. Lillicrap et al., “Continuous control with deep reinforcement learning,” *International Conference on Learning Representations (ICLR)*, 2016.

[24] T. Haarnoja, A. Zhou, P. Abbeel, and S. Levine, “Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor,” in *Proceedings of the 35th International Conference on Machine Learning*, 2018, pp. 1861–1870.

[25] J. Schulman, F. Wolski, P. Dhariwal, A. Radford, and O. Klimov, “Proximal policy optimization algorithms,” in *arXiv preprint arXiv:1707.06347*, 2017.