Read-Me

This project is a replication and methodological extension of 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).

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 those used in the main report (TD3 configuration). 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 **??**).

- `4_2.png`

  Baseline figure used for comparison with [1] (Figure **??**).

A.4 How to Reproduce

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

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 [2] (twin critics, target policy smoothing, delayed actor updates).

- Seed averaging follows reproducibility recommendations in [3] 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 **??**; changing them can materially alter convergence speed and volatility.

References

[1] R. Chen and Z. Zhang, *Deep Reinforcement Learning in Labor Market Simulations.* In *Proceedings of the IEEE Symposium on Computational Intelligence for Financial Engineering and Economics (CIFEr)*, 2025. DOI: 10.1109/CIFER64978.2025.10975741.

[2] 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)*, 2018.

[3] P. Henderson et al., *Deep Reinforcement Learning that Matters.* In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2018.