Impact of Heavy-Tailed Rewards on Exploration Strategies in Insurance Underwriting

Johann Bartels, Karsten Bruns
https://github.com/trololoJoojb/RL-Project-LUH-

Abstract

We examine how exploration shapes learning and solvency in underwriting with heavy-tailed losses. In a simulated insurance market with Pareto-tailed severities and latent regimes, we train DQNs using (i) fixed ε, (ii) linearly annealed ε, and (iii) rate-corrected, k-persistent ε-greedy ("ez-greedy"). Under equal budgets and multiple seeds, we compare capital paths, maximum drawdown, ruin probability, and cross-seed training stability. Ez-greedy reduces ruin and drawdowns while preserving returns, highlighting the importance of exploration under heavy tails.

1 Introduction

Insurance underwriting is a sequential decision-making process where an agent must decide whether to accept or reject incoming insurance contracts and at which premium level to offer them. In practice, insurers operate under uncertainty about future claims, which may exhibit heavy-tailed distributions due to rare but catastrophic losses. Such environments pose particular challenges for reinforcement learning (RL) agents: large negative outliers can drastically alter returns and mislead value estimates, while changing risk regimes require adaptable exploration.

In this work, we study exploration strategies in a simulated underwriting environment, where claim severities follow a Pareto distribution with varying risk regimes. We compare three ϵ-greedy-based approaches within a Deep Q-Network (DQN) framework:

1. **Fixed ϵ-greedy**, with a constant exploration probability.
2. **Annealed ϵ-greedy**, where exploration decreases linearly over time.
3. **Rate-corrected k-persistent ϵ-greedy** (*ez-greedy*), which initiates exploration phases with a given probability and repeats the same random action for k consecutive steps, adjusting the base ϵ to match the marginal exploration rate of other methods.

Our primary research question is: *Does rate-corrected k-persistent ϵ-greedy improve long-term returns and reduce bankruptcy risk in heavy-tailed insurance environments compared to fixed and annealed ϵ-greedy?*

2 Related Work

Dabney et al. [2020] propose *ez-greedy*, a temporally persistent variant of the standard ϵ-greedy strategy, in which a selected exploratory action is repeated for a duration drawn from a predefined (potentially heavy-tailed) distribution. This design mitigates step-to-step incoherent randomness and promotes greater diversity in visited states, while preserving the simplicity and stationarity of the original method. Empirical results demonstrate performance improvements across tabular, linear, and deep reinforcement learning settings[Dabney et al., 2020]. As a classical methodological baseline, the original Deep Q-Network [Mnih et al., 2013] employs a linearly decaying ϵ-greedy exploration

schedule. Within the insurance domain, Treetanthiploet et al. [2023] apply reinforcement learning to dynamic pricing in comparison websites, whereas Young et al. [2024] employ it for portfolio steering in insurance markets. Both studies operate in simulated environments with clearly defined state and reward structures, conceptually analogous to the simulated underwriting environment considered in our work.

3 Approach

Our study combines a custom heavy-tailed insurance underwriting environment with a Deep Q-Network (DQN) agent trained under three different ϵ-greedy exploration schedules. Below, we describe the environment design, agent architecture, and exploration methods.

3.1 Environment: `InsuranceEnvV2`

The environment models a stylized insurance underwriting process in discrete time steps. At each step, the agent receives an application characterized by:

- **Age of customer** (integer, normalized),
- **Region of customer** (one-hot encoded, 5 categories),
- **Risk score of customer** (continuous),
- **Capital** (current funds),
- **Outstanding liabilities** (sum of unpaid claims),
- **Risk regime** (hidden market condition indicator).

The agent selects one of five discrete actions: reject the application, or accept it at a premium factor $\in \{0.9, 1.0, 1.1, 1.2\}$ times a specified base premium. Accepted contracts may generate claims after a stochastic delay; claim severities follow a Pareto distribution with shape parameter $\alpha = 1.35$, scale $x_m = 1.1$, and are further scaled by regime-dependent multipliers. Claim probabilities depend on applicant features and regime, clipped to $p_{\text{claim}} \leq 0.8$.

The reward at each time step is the received premium minus any claims paid at that step. If capital falls below zero, the episode ends in bankruptcy, resulting in an additional penalty. The number of episodes is limited by a fixed horizon, optionally, all outstanding claims are settled at the end.

3.2 Agent: Deep Q-Network

The learning agent is a feedforward Deep Q-Network with configurable hidden layer sizes, ReLU activations, and a linear output layer producing one Q-value per action. We maintain both, an *Q*-network (for action selection and learning) and a *target* network (for stable temporal-difference targets), updated periodically. Training uses experience replay with a fixed-capacity buffer, uniform sampling, and an MSE loss between predicted Q-values and one-step bootstrap targets:

$$y_t = r_t + \gamma \max_{a'} Q_{\text{target}}(s_{t+1}, a') \tag{1}$$

where γ is the discount factor.
Parameters are optimized with Adam and random seeds are controlled for reproducibility.

3.3 Exploration Strategies

We compare three ϵ-greedy exploration schedules:

1. **Fixed** ϵ: constant exploration probability throughout the training process.
2. **Annealed** ϵ: ϵ decreases linearly from ϵ_{start} to ϵ_{end} over the total number of training steps.
3. **Rate-corrected** k**-persistent** ϵ**-greedy (ez-greedy)**: with probability ϵ_{base}, an *exploration phase* begins by sampling a random action, which is then repeated for k consecutive steps. The base rate is adjusted as:
$$\epsilon_{\text{base}} = \frac{\epsilon}{k - \epsilon \cdot (k-1)} \tag{2}$$
to match the marginal exploration rate ϵ of the other methods.

Ez-greedy introduces temporal persistence into exploration, potentially enabling more effective deviation from the greedy policy in heavy-tailed, non-stationary environments.

Algorithm 1 Rate-corrected k-persistent ϵ-greedy (*ez-greedy*)

Require: exploration rate ϵ, persistence length k

$\epsilon_{\text{base}} \leftarrow \frac{\epsilon}{k-\epsilon\cdot(k-1)}$
`repeat_counter` $\leftarrow 0$
`repeat_action` $\leftarrow$ `None`
function SELECTACTION(Q, s)
    if `repeat_counter` > 0 **then**
        `repeat_counter` $\leftarrow$ `repeat_counter` $- 1$
        return `repeat_action`
    else if Uniform$(0, 1) < \epsilon_{\text{base}}$ **then**
        `repeat_action` $\leftarrow$ **RandomAction**()
        `repeat_counter` $\leftarrow k - 1$
        return `repeat_action`
    else
        return $\arg\max_a Q(s, a)$
    end if
end function

4 Experiments

We evaluate the three exploration strategies (Fixed ϵ, Annealed ϵ, ez-greedy) in the `InsuranceEnvV2` environment under identical training budgets and hyperparameters.

4.1 Experimental Setup

For each exploration variant, we perform runs with 10 different random seeds:

- **Seeds:** `base_seed`, . . . , `base_seed` $+ 9$
- **Episodes:** fixed horizon per episode
- **Evaluation:** every N_{eval} training episodes, we run a greedy policy ($\epsilon = 0$) for M evaluation episodes.

The DQN hyperparameters, network architecture, replay buffer size, target network update frequency, and optimizer settings are kept constant across all runs, which are listed in our GitHub documentation.

4.2 Hyperparametersearch

We use a two-phase random search. In Phase 1 we sample **80** configurations and train for **20** episodes (horizon **500**). As selection metric we use the **tail mean** (mean over the last third of episodes) of the EZ variant. In Phase 2 we retrain the **Top-3** configurations *plus* one conservative safety configuration for **100** episodes (horizon **500**).

4.3 Resources

All experiments were executed on a 64-bit Windows system with an **AMD Ryzen 7 5800U with Radeon Graphics** (base 1.90 GHz) and **16 GB RAM**. A single training run (10 seeds) took approximately **3 hours**.

4.4 Results

We report learning dynamics and risk outcomes for the three exploration schedules (Fixed ϵ, Annealed ϵ, and rate-corrected k-persistent *ez-greedy*) under identical training budgets and seeds.

Exploration dynamics. As intended, annealed ϵ reduces exploration nearly linearly to zero over training, while *ez-greedy* and fixed-ϵ maintain a low but persistent number of exploratory steps throughout (Figure 3).

Acceptance behavior. Annealed ϵ exhibits a clear upward trend in acceptance rate from early to late training, with a variance uptick late in training. In contrast, *ez-greedy* and fixed-ϵ start with very high acceptance and show a mild downward drift over time (Figure 1). This pattern suggests that annealing transitions from wide early exploration to more selective underwriting, whereas the persistent schedules prioritize immediately "safe" choices but gradually become more conservative.

Pricing outcomes. Average premiums of accepted policies are highest for *ez-greedy* (and fixed-ϵ) early on and decline modestly over training. Annealed ϵ starts lower, increases mid-training, experiences a late dip, and partially recovers by the end (Figure 2). The policy exhibits a competitive yet risk-disciplined downward drift: initial safety margins are gradually reduced as information accumulates. Premiums decrease moderately and acceptance remains high, while solvency and loss targets are respected.

Bankruptcy risk. Training-time bankruptcy rates remain in the low single digits for all methods, displaying wave-like patterns consistent with periods of external disruptions. *Ez-greedy* tends to score the lowest rates, with annealed ϵ close and fixed-ϵ in between (Figure 4). In greedy evaluation, all variants achieve similarly low bankruptcy (roughly $\sim$1–3%), featuring a shared trough late in training and slightly more fluctuation for annealed ϵ (Figure 5).

Capital trajectories. Final capital trends upward for all strategies, with *ez-greedy* typically on top (annealed close behind), punctuated by synchronized drawdown windows that broaden the confidence bands (Figure 6). Minimum capital exhibits the same systemic dips across variants, confirming environment-driven heavy-tail shocks rather than strategy-specific failures (Figure 7).

Returns. Per-episode training returns oscillate around zero with rare but extreme negative spikes that align with the capital shock windows (Figure 8). Greedy-evaluation returns show a gradual upward drift. *Ez-greedy* frequently attains the highest top-end performance but with wider uncertainty bands, fixed-ϵ tracks closely, and annealed ϵ is slightly lower and is also slightly decreasing over time. (Figure 9).

Summary. Overall, all schedules learn profitable policies with low insolvency under greedy evaluation. *Ez-greedy* offers the highest capital levels and top-end returns but shows a competitive yet risk-disciplined downward drift with mild acceptance erosion over time. Annealed ϵ steadily improves acceptance and mid-training premiums but is more sensitive late in training. fixed-ϵ on the other hand, is competitive yet does not dominate on any dimension.

5 Discussion

Our results indicate a clear return–risk trade-off between temporally persistent and annealed exploration under heavy-tailed underwriting risk. The rate-corrected, k-persistent scheme (*ez-greedy*) more frequently reaches higher final capital levels and the upper quantiles of greedy-evaluation returns, but it does so with wider uncertainty bands, slightly more volatile learning, and a competitive yet risk-disciplined downward drift in pricing. That drift coincides with a mild decline in the acceptance rate over training, suggesting that the agent under *ez-greedy* increasingly prioritizes margins to buffer rare, large losses. In contrast, annealed-ϵ exhibits a textbook "explore-then-exploit" pattern: exploration steadily declines, acceptance rates rise over time, and capital trajectories remain comparatively smooth; however, once exploration has nearly vanished, late-stage regime shifts can produce sharper fluctuations. The fixed-ϵ baseline remains competitive but is less adaptive to regime changes and does not dominate any core metric across the entire training horizon.

Across all strategies we observe synchronized shock windows: simultaneous dips in minimum capital, rare extreme negative per-episode returns, and wave-like patterns in bankruptcy rates. These coincidences point to environment-driven tail events (Pareto losses and exogenous regime changes) as the primary driver of risk. The exploration schedule modulates robustness around these shocks

but cannot eliminate them. Practically, *ez-greedy* helps the policy escape short-sighted local optima through coherent bursts of random actions, thereby unlocking top-end performance, whereas annealed-ϵ is preferable when smooth capital paths and lower insolvency risk are the main objectives. Choosing between them should therefore reflect institutional priorities (growth and upside versus solvency and stability) and the tolerance for variance during shock periods.

Practical implications

For *maximum upside* (capital/returns) under heavy tails, *ez-greedy* is preferable, while for *robustness* with steadier acceptance and premiums, annealed ϵ is attractive. Fixed-ϵ is serviceable where simplicity and immediate stability matter, but it rarely beats the other two across metrics.

Limitations and future work. Findings are based on 10 seeds with wide confidence bands. Adding seeds and sensitivity studies (ez's k, target marginal ϵ, Annealed's floor) would strengthen evidence. Also, different environment configurations (action types or losses) could be further explored and investigating more sophisticated risk models beyond the Pareto distribution for claim sizes could better capture tail-risk behavior and enhance the robustness of policy evaluation. Furthermore, additional and more complex customer features or even dynamic profiles that evolve over time would allow for a richer and more realistic representation of market heterogeneity. To conclude, stabilizers such as Double-DQN, prioritized replay, and soft target updates could further reduce variance—especially for ez—and are promising extensions.

6 Figures

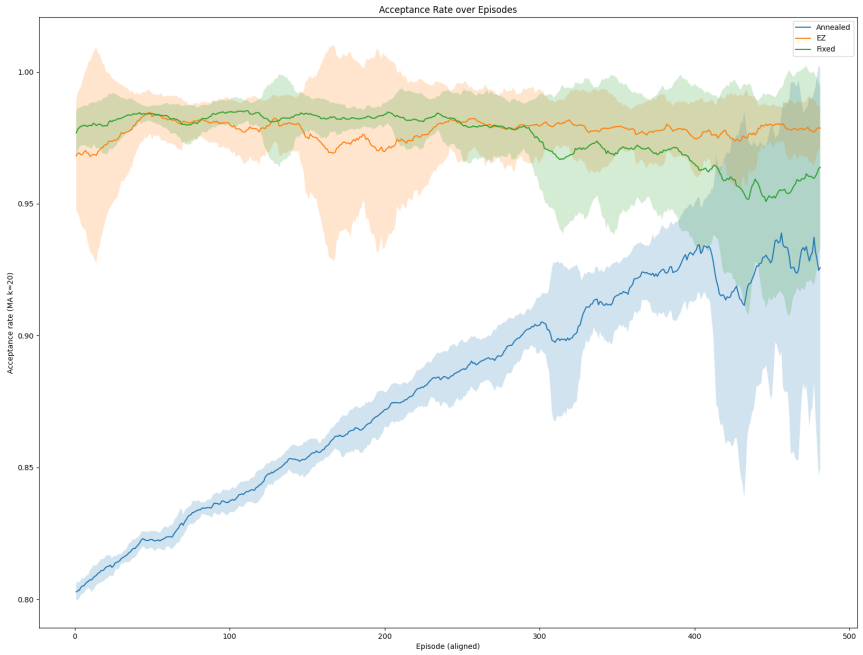

Figure 1: Acceptance rate over episodes (mean $\pm$ std across seeds).

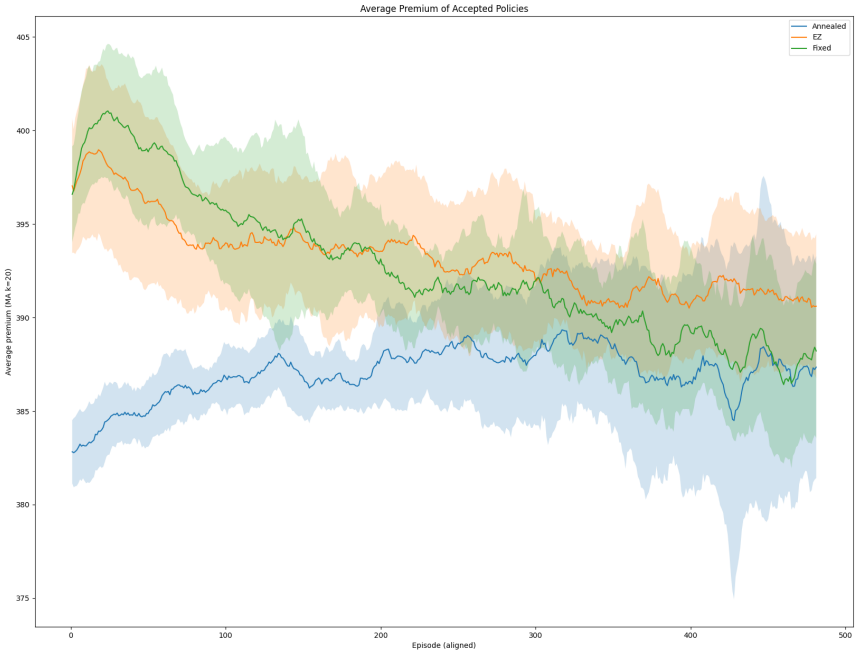

Figure 2: Average premium of accepted policies during training.

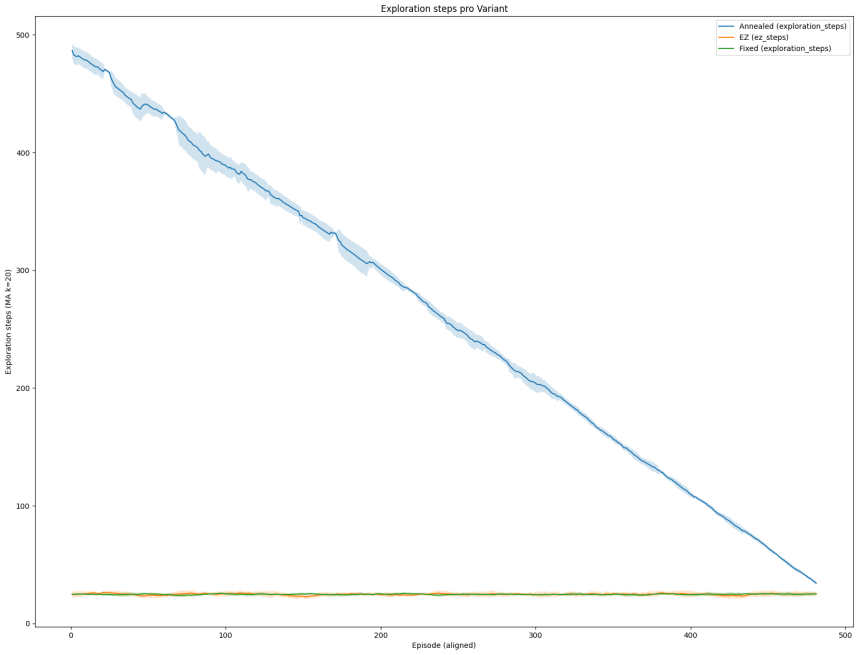

Figure 3: Exploration steps per episode for each strategy.

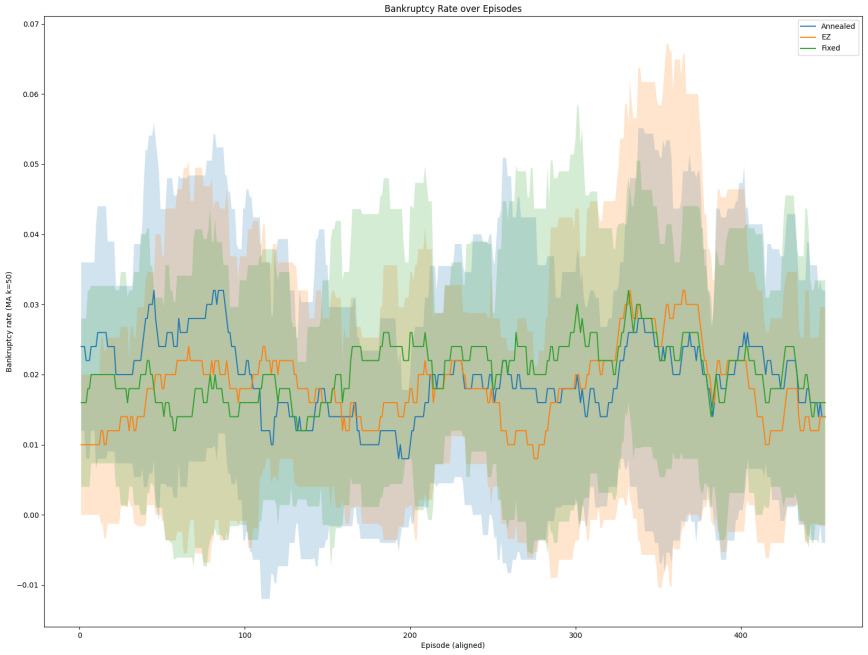

Figure 4: Bankruptcy rate over episodes (training).

Figure 5: Bankruptcy rate vs. training progress (greedy evaluation).

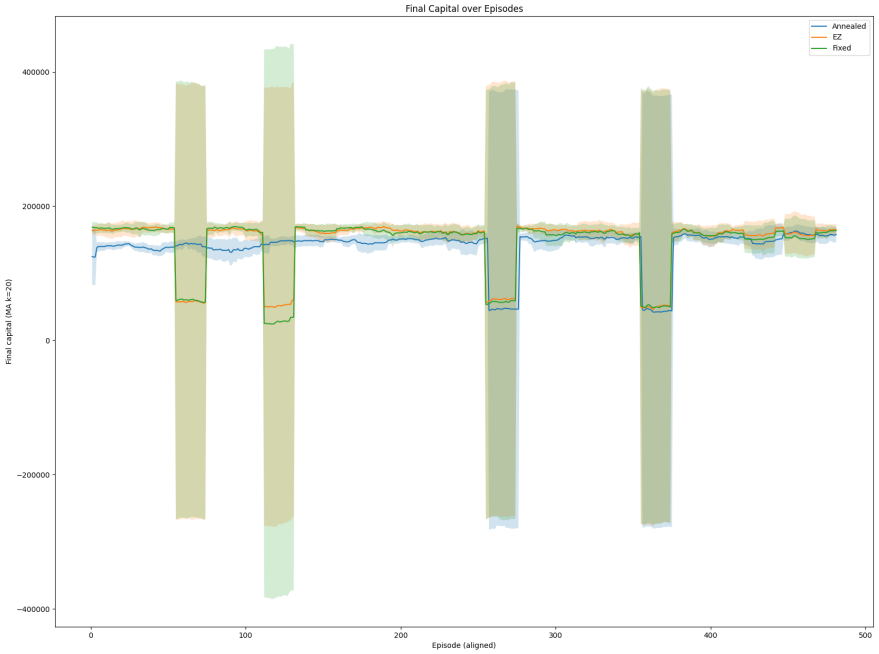

Figure 6: Final capital over episodes (training).

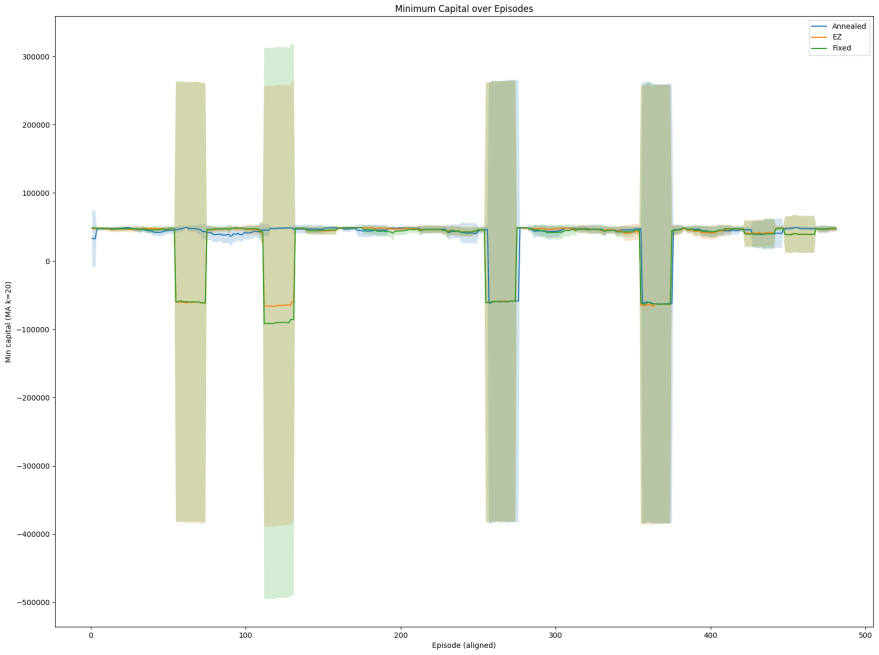

Figure 7: Minimum capital over episodes (training).

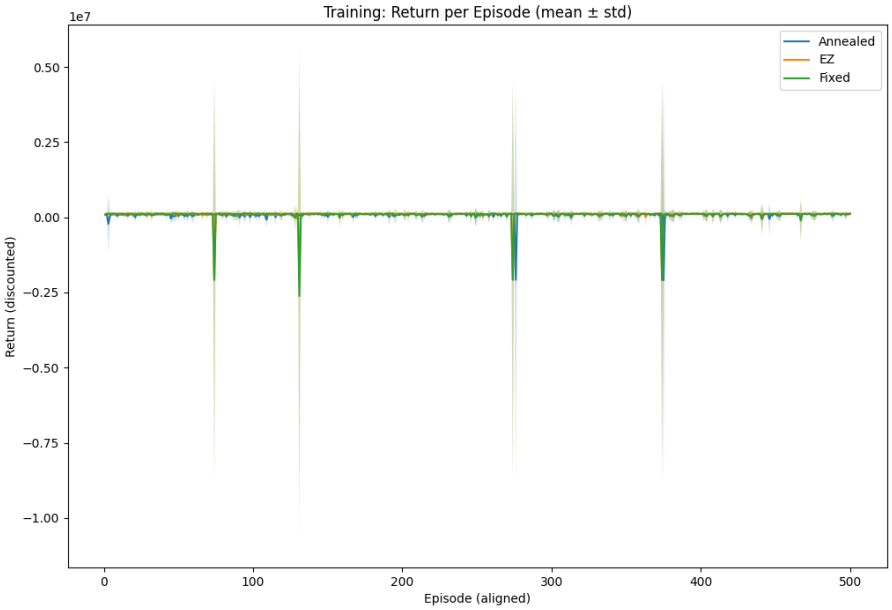

Figure 8: Return per episode during training (discounted).

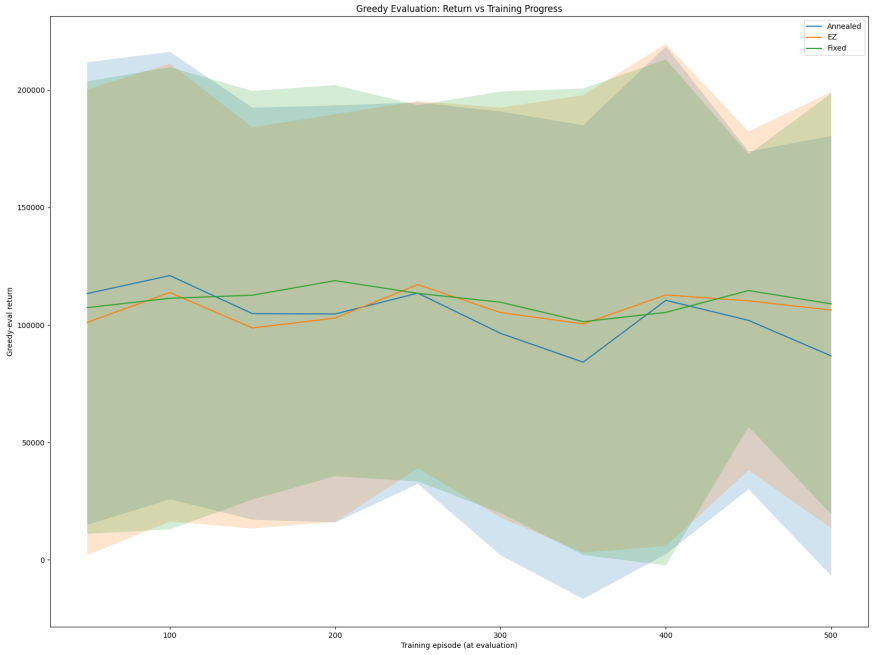

Figure 9: Greedy-evaluation return vs. training progress.

References

Will Dabney, Georg Ostrovski, and André Barreto. Temporally-extended ϵ-greedy exploration. *arXiv preprint arXiv:2006.01782*, 2020.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Playing atari with deep reinforcement learning. *arXiv preprint arXiv:1312.5602*, 2013.

Tanut Treetanthiploet, Bhanu Prakash, and Narendra Nath. Insurance pricing on price comparison websites via reinforcement learning. *arXiv preprint arXiv:2308.06935*, 2023.

Edward J. Young, Alistair Rogers, Will Clark, and Mike Giles. Reinforcement learning applied to insurance portfolio pursuit. *arXiv preprint arXiv:2408.00713*, 2024.

Checklist

This checklist is not mandatory, but can help you set up your project in a reproducible manner. Options for filling it out are: [Yes] [No] [NA]

1. General points:
   (a) Do the main claims made in the abstract and introduction accurately reflect your contributions and scope? [Yes]
   (b) Did you cite all relevant related work? [Yes]
   (c) Did you describe the limitations of your work? [Yes]
   (d) Did you include a discussion of future work? [Yes]
2. If you are including theoretical results...
   (a) Did you state the full set of assumptions of all theoretical results? [NA]
   (b) Did you include complete proofs of all theoretical results? [NA]
3. If you ran experiments (e.g. for benchmarks)...
   (a) Did you include the code, data, and instructions needed to reproduce the main experimental results (either in the supplemental material or as a URL)? [Yes]
   (b) Did you specify all the training details (e.g., data splits, hyperparameters, how they were chosen)? [Yes]
   (c) Did you run at least 10 repetitions of your method? [Yes]
   (d) Did you report error bars (e.g., with respect to the random seed after running experiments multiple times)? [Yes]
   (e) Did you include the total amount of compute and the type of resources used (e.g., type of GPUs, internal cluster, or cloud provider)? [Yes]
4. If you are using existing assets (e.g., code, data, models) or curating/releasing new assets...
   (a) If your work uses existing assets, did you cite the creators? [Yes]
   (b) Did you make sure the license of the assets permits usage? [Yes]
   (c) Did you reference the assets directly within your code and repository? [Yes]