RL Project Report:

Exploring Value Function Transfer Between On-Policy and Off-Policy Methods in Tabular Gridworld

Can Kocak and Paul Steinbrink (rl_ckps)
https://github.com/cako2025/rl_ckps_final_project/

Abstract

We investigate tabular value function transfer between on-policy (SARSA) and off-policy (Q-Learning) reinforcement learning in the stochastic FrozenLake environment. By systematically varying both source and target algorithms, as well as exploration strategies (ϵ-greedy, softmax), we evaluate how transferred Q-values affect learning speed, final performance, and stability. Our experiments reveal that matching the pre-training agent's algorithm and exploration method with the transfer agent yields the greatest and most reliable benefits, significantly boosting sample efficiency. Conversely, cross-paradigm transfers provide little advantage or can even be detrimental. Statistical analysis confirms these trends, underscoring the sensitivity of RL transfer learning effectiveness to algorithmic and exploration congruence.

1 Introduction

Transfer learning in reinforcement learning (RL) seeks to accelerate learning by leveraging knowledge from previously trained policies. In tabular RL, transferring value functions between different algorithms offers a controlled setting to study this process, especially when crossing the boundary between off-policy and on-policy paradigms, which differ fundamentally in how they collect and utilize experience.

This work investigates whether tabular value functions learned by one RL algorithm can improve learning when transferred to another, and how exploration strategies influence this transfer. We focus on Q-Learning (off-policy) and SARSA (on-policy) in the FrozenLake environment from Gymnasium - a discrete, stochastic, episodic task with a small but variable state space that enables analysis of learning stability under stochastic returns and variable episode lengths. Our research questions are:

- **RQ1:** Can value functions be effectively transferred between off-policy and on-policy tabular RL methods, and vice versa?
- **RQ2:** How does the choice of exploration strategy (ϵ-greedy vs. softmax) influence the success and dynamics of such transfer?

We analyze transferred policies with respect to cumulative return, episode length, and temporal difference (TD) error, comparing their impact to agents trained from scratch. This study provides insights into how algorithmic paradigms and exploration strategies interact with transferred knowledge in tabular domains, offering practical guidance for initialization strategies in small-scale RL tasks.

2 Related Work

Transfer learning in reinforcement learning has been studied as a way to improve sample efficiency by reusing knowledge across tasks or agents [Lazaric, 2012, Zhu et al., 2023]. In the tabular setting, value function transfer offers interpretability and low computational overhead, and prior work has examined its impact on convergence and stability in small-scale MDPs [Zhu et al., 2023].

On-policy and off-policy methods, such as SARSA and Q-Learning, differ in how they learn from experience and exhibit distinct convergence behaviors influenced by exploration strategy and reward structure [Sutton and Barto, 2018]. These differences are critical when initializing one algorithm with the value function of another.

Exploration strategies impact learning dynamics and performance in RL. Tokic and Palm [2011] introduced VDBE-Softmax, an adaptive method that triggers exploration based on uncertainty in value estimates, outperforming standard ϵ-greedy and softmax in both on-policy and off-policy settings. Building on this, our work examines how ϵ-greedy and softmax exploration interact with transferred value functions in a stochastic tabular environment.

3 Approach

Our method investigates the impact of value function transfer between on-policy and off-policy tabular RL algorithms in the Frozenlake environment. We train an agent using either SARSA or Q-Learning for a fixed number of episodes, then transfer the learned value function $Q(s, a)$ to the other algorithm. During both phases, we vary the exploration strategy between ϵ-greedy and softmax.

All hyper-parameters were tuned with SMAC [Lindauer et al., 2022] on a deterministic version of the FrozenLake environment to reduce noise and speed up the search. After selecting the best configuration, we conducted all transfer experiments on the stochastic version.

Specifically, we implement the following experimental variants:

- **Q $\rightarrow$ SARSA:** Pretrain Q-Learning for N episodes, transfer $Q(s, a)$ to SARSA, and continue training.
- **SARSA $\rightarrow$ Q:** Pretrain SARSA for N episodes, transfer $Q(s, a)$ to Q-Learning, and continue training.
- For each, we evaluate all combinations of exploration strategies (*softmax* $\rightarrow$ *softmax*, *softmax* $\rightarrow$ *ϵ-greedy*, etc.).

The pseudocode below summarizes the full transfer procedure:

Algorithm 1 Value Function Transfer Across RL Algorithms

```
Require: Env $e$, Algo1 $A_1$, Algo2 $A_2$, exploration modes $E_1$, $E_2$, switch time $N$
  Initialize $Q(s, a) \leftarrow 0$
  for $i = 1$ to $N$ do
      Train $A_1$ using $Q(s, a)$ and exploration $E_1$
  end for                                              ▷ Transfer: reuse $Q(s, a)$
  for $i = N + 1$ to total episodes do
      Train $A_2$ using $Q(s, a)$ and exploration $E_2$
  end for
  return Final $Q(s, a)$, performance metrics
```

The action selection function π depends on the exploration method:

$$\pi(a|s) = \begin{cases} \text{softmax}(Q(s,a), \tau) & \text{if softmax} \\ \epsilon\text{-greedy}(Q(s,a), \epsilon) & \text{if } \epsilon\text{-greedy} \end{cases} \tag{1}$$

The update rule depends on the algorithm being used:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha \left[r_t + \gamma \cdot \left(\begin{cases} Q(s_{t+1}, a_{t+1}) & \text{SARSA} \\ \max_a Q(s_{t+1}, a) & \text{Q-Learning} \end{cases} \right) - Q(s_t, a_t) \right] \tag{2}$$

In all transfer experiments, the number of episodes for the transfer agent is equal to the number of episodes for the pre-trained agent, which is fixed at 5,000. This setup allows us to isolate the effects of algorithmic transfer and exploration strategy changes on learning performance in a reproducible, interpretable environment.

4 Experiments

Each figure presents three subplots with a logarithmically scaled x-axis for the episodes. Shaded regions illustrate run-to-run variance. To assess statistical differences, Levene's tests are used to compare variances, while Welch's or Student's t-tests evaluate differences in means at both the beginning and end of the transfer window (window size: 500). The area under the learning curve (AUC) provides a summary of overall performance, where higher AUC is favorable for return, and lower AUC is preferable for episode length and training error.

All tables showing trends include mean (W) and slope (t) for early and late training, with p-values from Levene's and t-tests ($\alpha = 0.05$).

Q-learning ϵ-Greedy

The plotted curves in Figure 1 reveal that Q-Learning using ϵ-greedy exploration pre-trained with Q-Learning softmax attains the highest cumulative return but exhibits longer episodes and increased training error later in training, whereas pre-training with their own policy leads to shorter episodes and lower error, though with more moderate returns.

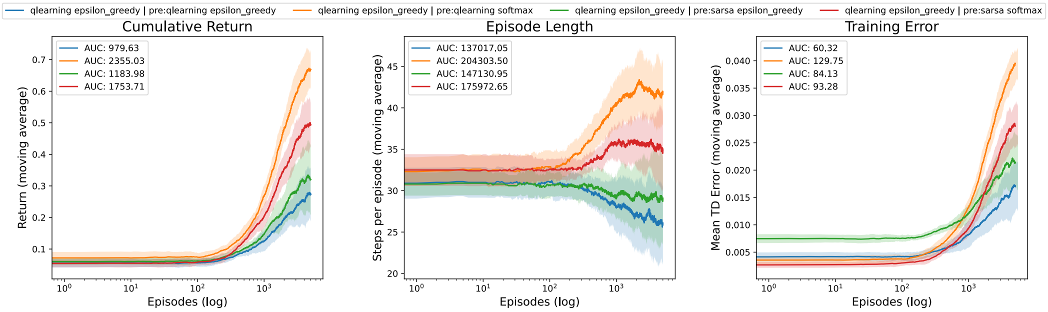

Figure 1: Learning performance of Q-Learning (ϵ-greedy) after transfer from four different pre-trained agents. Subplots show cumulative return, episode length, and TD error (log-scaled x-axis).

As shown in Tables 1 - 3, Q-Learning softmax pre-training leads to high early returns and long episodes, but steep declines later in both return and TD error, indicating instability. SARSA ϵ-greedy shows more consistent trends across all metrics, with shorter episodes and faster error reduction, suggesting more stable transfer performance.

Table 1: Return trends with statistical tests.

Q-Learning ϵ-greedy	W_{start}	p	t_{start}	p	W_{end}	p	t_{end}	p
vs. Q-Learning softmax	285.115	0.000	27.102	0.000	0.015	0.903	-3409.440	0.000
vs. SARSA ϵ-greedy	16.113	0.000	-9.413	0.000	292.721	0.000	-249.810	0.000
vs. SARSA softmax	148.214	0.000	-9.773	0.000	268.269	0.000	-1229.166	0.000

Table 2: Episode-Length trends with statistical tests.

Q-Learning ϵ-greedy	W_{start}	p	t_{start}	p	W_{end}	p	t_{end}	p
vs. Q-Learning softmax	688.588	0.000	-56.569	0.000	8.194	0.004	-1348.167	0.000
vs. SARSA ϵ-greedy	410.561	0.000	-7.892	0.000	48.814	0.000	-315.100	0.000
vs. SARSA softmax	3.444	0.064	-86.184	0.000	61.134	0.000	-699.879	0.000

Table 3: TD-Error trends with statistical tests.

Q-Learning ϵ-greedy	W_{start}	p	t_{start}	p	W_{end}	p	t_{end}	p
vs. Q-Learning softmax	330.994	0.000	-4.630	0.000	181.266	0.000	-1243.470	0.000
vs. SARSA ϵ-greedy	54.125	0.000	-80.367	0.000	26.809	0.000	-342.330	0.000
vs. SARSA softmax	97.196	0.000	17.177	0.000	0.073	0.787	-957.037	0.000

Q-learning Softmax

The plotted curves in Figure 2 reveal that Q-Learning using softmax exploration pre-trained with SARSA softmax achieves the lowest training error and shorter episodes, but yields more moderate cumulative returns. In contrast, pre-training with SARSA ϵ-greedy policy leads to the highest cumulative return, although the cost of increased training error and slightly longer episodes as training progresses.

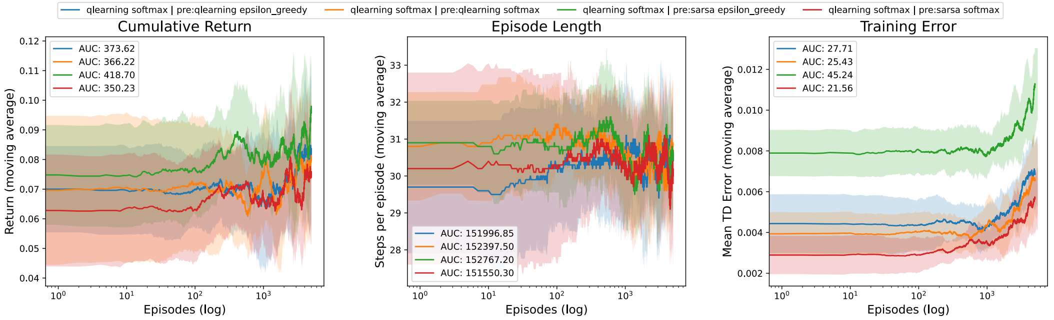

Figure 2: Learning performance of Q-Learning (softmax) after transfer from four different pre-trained agents. Subplots show cumulative return, episode length, and TD error (log-scaled x-axis).

As shown in Tables 4 - 6, SARSA ϵ-greedy pre-training yielded the highest cumulative return but also higher training error and less efficient episode reduction. In contrast, SARSA softmax led to the lowest TD error and shortest episodes, albeit with more moderate returns. Transfer from Q-Learning ϵ-greedy showed balanced performance across all metrics, with stable learning and consistent improvement.

Table 4: Return trends with statistical tests.

Q-Learning softmax	W_{start}	p	t_{start}	p	W_{end}	p	t_{end}	p
vs. Q-Learning ϵ-greedy	64.782	0.000	-2.421	0.016	336.871	0.000	-17.263	0.000
vs. SARSA ϵ-greedy	597.076	0.000	-54.957	0.000	818.851	0.000	-43.685	0.000
vs. SARSA softmax	135.182	0.000	12.681	0.000	53.997	0.000	41.898	0.000

Table 5: Episode-Length trends with statistical tests.

Q-Learning softmax	W_{start}	p	t_{start}	p	W_{end}	p	t_{end}	p
vs. Q-Learning ϵ-greedy	15.485	0.000	65.361	0.000	66.185	0.000	17.792	0.000
vs. SARSA ϵ-greedy	90.394	0.000	3.672	0.000	2.949	0.086	22.230	0.000
vs. SARSA softmax	110.243	0.000	42.954	0.000	134.890	0.000	57.380	0.000

Table 6: TD-Error trends with statistical tests.

Q-Learning softmax	W_{start}	p	t_{start}	p	W_{end}	p	t_{end}	p
vs. Q-Learning ϵ-greedy	31.633	0.000	-111.025	0.000	609.202	0.000	-45.272	0.000
vs. SARSA ϵ-greedy	29.831	0.000	-863.147	0.000	1813.978	0.000	-191.178	0.000
vs. SARSA softmax	437.132	0.000	111.961	0.000	1447.504	0.000	160.272	0.000

SARSA ϵ-Greedy

The plotted curves in Figure 3 reveal that SARSA using ϵ-greedy exploration pre-trained with Q-Learning softmax attains the highest cumulative return but results in longer episodes and increased training error later in training. By contrast, pre-training with Q-Learning using the ϵ-greedy policy yields shortest episodes and lowest training error, although with the most less cumulative returns.

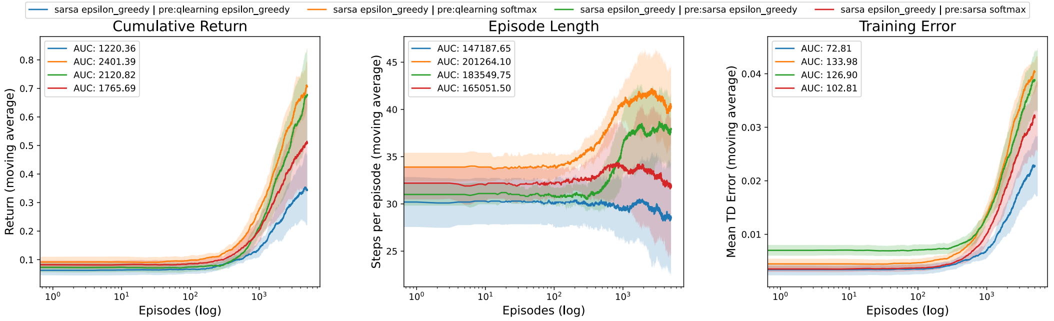

Figure 3: Learning performance of Q-Learning (ϵ-greedy) after transfer from four different pre-trained agents. Subplots show cumulative return, episode length, and TD error (log-scaled x-axis).

As shown in Tables 7 - 9, pre-training SARSA ϵ-greedy with Q-Learning softmax led to the highest returns but at the cost of longer episodes and elevated training error. In contrast, Q-Learning ϵ-greedy pre-training resulted in the shortest episodes and lowest TD error, though with lower returns. SARSA softmax showed weak and unstable learning across all metrics, with minimal return gains and rising error.

Table 7: Return trends with statistical tests.

SARSA ϵ-greedy	W_{start}	p	t_{start}	p	W_{end}	p	t_{end}	p
vs. Q-Learning ϵ-greedy	23.607	0.000	6.218	0.000	71.767	0.000	1010.035	0.000
vs. Q-Learning softmax	129.543	0.000	-32.634	0.000	448.489	0.000	-40.190	0.000
vs. SARSA softmax	36.087	0.000	-18.897	0.000	0.294	0.588	427.102	0.000

Table 8: Episode-Length trends with statistical tests.

SARSA ϵ-greedy	W_{start}	p	t_{start}	p	W_{end}	p	t_{end}	p
vs. Q-Learning ϵ-greedy	81.793	0.000	75.555	0.000	3.070	0.080	940.535	0.000
vs. Q-Learning softmax	1111.742	0.000	-88.433	0.000	106.651	0.000	-203.926	0.000
vs. SARSA softmax	83.345	0.000	-84.035	0.000	0.513	0.474	622.492	0.000

Table 9: TD-Error trends with statistical tests.

SARSA ϵ-greedy	W_{start}	p	t_{start}	p	W_{end}	p	t_{end}	p
vs. Q-Learning ϵ-greedy	23.410	0.000	97.877	0.000	4.050	0.044	1084.025	0.000
vs. Q-Learning softmax	171.746	0.000	38.397	0.000	512.265	0.000	-49.123	0.000
vs. SARSA softmax	64.739	0.000	80.593	0.000	363.550	0.000	238.393	0.000

SARSA Softmax

The plotted curves in Figure 4 reveal that SARSA using softmax exploration pre-trained with SARSA softmax achieves the lowest training error and shorter episode lengths, though with more moderate cumulative returns. In contrast, pre-training with SARSA ϵ-greedy results in the highest cumulative return but at the expense of increased training error and slightly longer episodes later in training.

As shown in Tables 10 - 12, SARSA softmax pre-training led to the lowest training error and short episode lengths, but moderate cumulative returns. In contrast, SARSA ϵ-greedy achieved the highest early returns, though with increasing error and less consistent episode behavior. Q-Learning-based

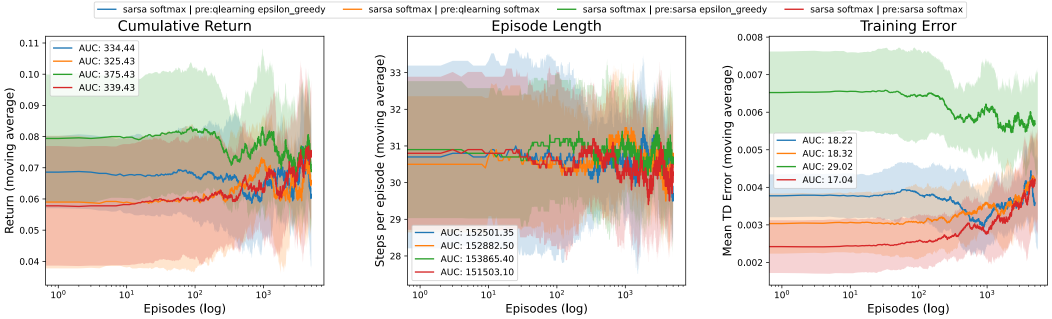

Figure 4: Learning performance of Q-Learning (ϵ-greedy) after transfer from four different pre-trained agents. Subplots show cumulative return, episode length, and TD error (log-scaled x-axis).

pre-training showed gradual improvement across all metrics, indicating stable and adaptive transfer despite weaker early performance.

Table 10: Return trends with statistical tests.

SARSA softmax	W_{start}	p	t_{start}	p	W_{end}	p	t_{end}	p
vs. Q-Learning ϵ-greedy	0.507	0.476	-72.068	0.000	330.860	0.000	54.248	0.000
vs. Q-Learning softmax	116.336	0.000	-7.584	0.000	160.553	0.000	59.210	0.000
vs. SARSA ϵ-greedy	1255.205	0.000	-89.177	0.000	306.407	0.000	-5.371	0.000

Table 11: Episode-Length trends with statistical tests.

SARSA softmax	W_{start}	p	t_{start}	p	W_{end}	p	t_{end}	p
with Q-Learning ϵ-greedy	23.869	0.000	13.824	0.000	367.207	0.000	5.707	0.000
with Q-Learning softmax	143.340	0.000	14.858	0.000	0.315	0.575	-50.670	0.000
with SARSA ϵ-greedy	8.332	0.004	-17.259	0.000	73.161	0.000	-48.424	0.000

Table 12: TD-Error trends with statistical tests.

SARSA softmax	W_{start}	p	t_{start}	p	W_{end}	p	t_{end}	p
with Q-Learning ϵ-greedy	7.057	0.008	-105.711	0.000	529.147	0.000	38.615	0.000
with Q-Learning softmax	35.415	0.000	-65.441	0.000	93.787	0.000	-36.624	0.000
with SARSA ϵ-greedy	737.425	0.000	-244.585	0.000	4.081	0.044	-497.654	0.000

5 Discussion

In this work, we explored the efficacy of tabular value function transfer between on-policy (SARSA) and off-policy (Q-Learning) reinforcement learning algorithms within the stochastic FrozenLake environment. Our primary objective was to address the research questions mentioned in section 1 concerning value function transfer and exploration strategies.

Our results demonstrate that value function transfer is most effective when the pretraining and target agents share both the same algorithm and exploration strategy (e.g., SARSA softmax $\rightarrow$ SARSA softmax). Such alignment leads to faster convergence, reduced TD error, and more stable episode lengths, confirming the benefits of intra-paradigm transfer and affirmatively answering RQ1 in this context. Conversely, transfers across different algorithms or with mismatched exploration methods frequently caused unstable or degraded performance. For instance, transferring from SARSA softmax to Q-Learning ϵ-greedy resulted in increased TD errors and less effective learning, indicating that differences in learning dynamics and action selection strategies can undermine transfer success. Hence, cross-paradigm transfers should be approached cautiously, offering only limited or conditional benefits.

Regarding RQ2, our findings clearly show that exploration strategy congruence plays a crucial role in transfer effectiveness. Matching exploration methods between source and target agents consistently correlated with improved learning outcomes, while mismatches often introduced volatility, longer episode durations, and elevated TD errors. Although softmax pre-training sometimes yielded higher early returns, these gains were often not sustained without consistent exploration alignment, emphasizing the sensitivity of transfer to the exploration approach.

Despite these insights, several limitations remain. Our study was restricted to a single, relatively simple environment, and only two algorithms and two exploration strategies were evaluated. This scope may limit the generalizability of our findings. Future work could extend this analysis to more complex environments and a broader set of RL methods and exploration policies. Additionally, investigating state-wise transferability, adaptive transfer mechanisms, or hybrid exploration strategies could improve robustness and broaden the applicability of value function transfer across diverse RL paradigms.

References

Alessandro Lazaric. Transfer in reinforcement learning: A framework and a survey. In Marco A. Wiering and Martijn van Otterlo, editors, *Reinforcement Learning*, volume 12 of *Adaptation, Learning, and Optimization*, pages 143–173. Springer, 2012. doi: 10.1007/978-3-642-27645-3_5. URL `https://doi.org/10.1007/978-3-642-27645-3_5`.

Marius Lindauer, Katharina Eggensperger, Matthias Feurer, André Biedenkapp, Difan Deng, Carolin Benjamins, Tim Ruhkopf, René Sass, and Frank Hutter. Smac3: A versatile bayesian optimization package for hyperparameter optimization. *Journal of Machine Learning Research*, 23(54):1–9, 2022. URL `http://jmlr.org/papers/v23/21-0888.html`.

Richard S. Sutton and Andrew G. Barto. *Reinforcement learning - an introduction, 2nd Edition*. MIT Press, 2018. URL `http://www.incompleteideas.net/book/the-book-2nd.html`.

Michel Tokic and Günther Palm. Value-difference based exploration: Adaptive control between epsilon-greedy and softmax. In Joscha Bach and Stefan Edelkamp, editors, *KI 2011: Advances in Artificial Intelligence, 34th Annual German Conference on AI, Berlin, Germany, October 4-7,2011. Proceedings*, volume 7006 of *Lecture Notes in Computer Science*, pages 335–346. Springer, 2011. doi: 10.1007/978-3-642-24455-1_33. URL `https://doi.org/10.1007/978-3-642-24455-1_33`.

Zhuangdi Zhu, Kaixiang Lin, Anil K. Jain, and Jiayu Zhou. Transfer learning in deep reinforcement learning: A survey. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(11):13344–13362, 2023. doi: 10.1109/TPAMI.2023.3292075. URL `https://doi.org/10.1109/TPAMI.2023.3292075`.