RL Project Proposal:
Influence of Transfer Learning on Performance

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

1 Motivation

Transfer learning has shown promise in accelerating learning by leveraging prior knowledge, but its impact on the performance of reinforcement learning agents in stochastic environments like Blackjack remains unclear. Understanding whether transfer learning improves or changes the agent's win rate is essential for designing effective training strategies.

This motivation leads us to the hypothesis:

- **H0:** Transfer learning (training an agent after a previously trained agent) does not significantly affect the average win rate of the second agent compared to solo training without transfer.

- **H1:** Transfer learning leads to a significant change in the average win rate of the second agent compared to solo training.

2 Related Topics

RL-Algorithm: Q-Learning (off-policy) and SARSA (on-policy).
exploration-strategy: epsilon-greedy and softmax.
topic: transfer-learning

3 Idea

We first train a policy π_{pre} using a standard reinforcement learning algorithm. This policy is then transferred to a second agent π_{transfer}, while a separate agent π_{scratch} is trained from scratch using the same algorithm and environment. We then compare the expected performance of both agents to determine the effect of transfer learning.

The hypothesis is formalized as:

$$\pi \in \Pi, \pi : \mathcal{S} \mapsto \mathcal{A}, \quad \mathbb{E}[R(\pi_{\text{transfer}})] \neq \mathbb{E}[R(\pi_{\text{scratch}})] \tag{1}$$

where $\mathbb{E}[R(\pi)]$ denotes the expected return of policy π.

Algorithm 1 Transfer Learning vs. Solo Training

Require: environment e, RL algorithm A
  Train first agent π_{pre} on e
  Initialize second agents:
    Option 1 (Transfer): $\pi_{\text{transfer}} \leftarrow \pi_{\text{pre}}$
    Option 2 (Solo): randomly initialize π_{scratch}
  while not converged **do**
    Train π_{transfer} on e using A
    Train π_{scratch} on e using A
  end while
  Compute $\mathbb{E}[R(\pi_{\text{transfer}})]$ and $\mathbb{E}[R(\pi_{\text{scratch}})]$
  return comparison of both performance values

4 Experiments

We want to know roughly what you are planning to show in terms of experiments.

Environments & Metrics
ENV: Blackjack-v1 (Gymnasium - Toy Text)
Metrics: average win rate of the agent over multiple episodes, learning speed and convergence behavior.

Experimental Scope
How many experiments are you running? Include seeds, hyperparameter optimization, different environments, ablations, etc. here.

Estimated Computational Load
Training times vary by algorithm but are expected to be moderate due to the simplicity of the Blackjack environment. Each training run should take between several minutes to an hour on a standard CPU/GPU. Experiments will be automated via scripts and run on a local machine. Overall, the project is designed to be computationally feasible within a few days to a week.

5 Timeline

Estimated timeline for our project includes:

- Research: 2 days — Literature review on transfer learning and RL algorithms in Blackjack.
- Implementation: 2 days — Implement training pipelines for the RL algorithms and transfer learning setups.
- Experiments: 4 days — Run experiments across different algorithms, seeds, and settings.
- Analysis: 2 days — Statistical evaluation of results, hypothesis testing, and interpretation.
- Reporting: 1.5 days — Writing the final report, preparing visualizations, and presentation.