# Grid World Transfer Learning

**Tyler McDonnell**
Department of Computer Science
The University of Texas at Austin
tmcdonnell@utexas.edu

## Introduction

Reinforce Learning (RL) agents learn by interacting with the environment. Whether the world they are interacting with is virtual or physical, there is always a cost to learning in this manner: e.g., compute time in the virtual case, and any number of costs when interacting online with the real world. Transfer Learning (TL) involves generalizing knowledge across data drawn from different distributions or domains, with the goal of reducing the amount of data required to learn a new or related task.

There are many different flavors of TL techniques. In general, these techniques exist on a spectrum in which one trades computational requirements and/or data complexity for flexibility of transfer. For this report, I focus on one of the simplest, but least flexible, varieties of TL, in which we copy the action-value function directly from a source to a target task. The action-value function is then used as a starting point for learning in the target domain. This approach assumes a high degree of similarity between tasks. In my evaluation, both tasks share the exact same action space, dimensions of state space, and agent learning algorithms.

In this report, my goals are to answer the following:

1. How does direct action-value copy TL work in practice?
2. How does domain size affect TL?
3. How do small variations in the environment affect TL?

To answer these questions, I apply direct action-value copying TL in the Grid World domain, in which an agent learns to navigate through an $MxN$ world. The Grid World domain is interesting because it is very simple and easy to analyze policies. Furthermore, the lessons learned from it can be generalized to much more complex or difficult navigational tasks that abound in more practical real-world RL.

## Grid World

The experiments in this report are carried out in the Grid World domain. Each Grid World can be described as an $MxN$ array of grid spaces, with a label for each grid space. The labels are: START, the space where the agent starts; GOAL, the space the agent is trying to reach; BLOCKED, a space that cannot be navigated by the agent; and *Pit*, an agent which causes the agent to lose.

The goal of a Grid World task is for the agent to move from the START space to the GOAL space as quickly as possible. I adopt a common Grid World formulation in which rewards are distributed as follows: +1 for reaching the GOAL state; -1 for entering a PIT; and -0.001 for taking an action which results in neither of the above.

I adopt an Episodic formulation, in which episodes begin at the START state and end when an agent reaches either the GOAL state or a PIT state. Additionally, for learning, I adopt a tabular Q-Learning approach for computing action values. Since I am interested mainly in the application of TL, I do not consider other learning approaches.

As mentioned, Grid World is a compelling case study for TL because it is easy to analyze and serves as a representative simplification of more complex navigational tasks. The experiments and Grid World variants in the experiments that follow were implemented on top of the RLPy framework (Geramifard12 et al. 2015).

## Methodology

This work explores one of the simplest and least flexible TL methods, the direct transfer of the action-value function from a source task to a target task. As previously mentioned, we assume that the source and target task share the exact same action space, dimensionality of state space, and agent learning algorithms. In each experiment that follows, a source task and target task grid world will be considered, each of identical dimensionality $MxN$. In each case, a Tabular Q-Learning agent, SOURCE is first trained in the source grid world for some number of steps. Then, a Tabular Q-Learning agent TARGETNOTRANSFER with an $\epsilon$-Greedy Policy and initial learning rate of 0.1 is trained on the target grid world. Finally, a third agent, TARGETTRANSFER is initialized using the action-values learned from SOURCE and then trained in the target grid world over the same number of steps. During training, an $\epsilon$-Greedy policy is used, with $\epsilon = 0.2$. During policy evaluation, the deterministic learned policy is used. The performance of TARGETNOTRANSFER and TARGETTRANSFER is then compared to evaluate TL.

## Domain Size

My first experiment attempts to characterize the utility of TL on Grid World domains of different sizes. My hypothesis was that TL would be more useful for speeding learning in large state spaces, which might otherwise take a large amount of computational overhead to learn. For these experiments, I randomly generated square grid worlds of increasing size, from 3x3 to 10x10. For each world, the START space was located at (0,0), and the GOAL space was located at $(N-1, N-1)$. I also randomly generated $N * N/16$ BLOCKED squares and $(N * N/8)$ PIT squares for each world. This decision was arbitrary and was meant to introduce sufficient obstacles without overly obstructing the state space. I then apply the TL experiment as described in **Methodology** and measure the performance of direct action-value transfer.

Table 1 shows the results in terms of **transfer ratio**, the total reward accumulated by the agent with transferred knowledge divided by the total reward accumulated by the agent with no transferred knowledge. Unfortunately, this characterization did not progress far. First of all, I realized that the locations I chose for the START and GOAL states were somewhat adversarial in terms of computational complexity: the states are as far possible from one another. This made learning increasingly large grid worlds extremely expensive. Moreover, it quickly became clear that domain size was not clearly correlated with positive or negative transfer: the results from the 3x3, 4x4, and 5x5 grids were all over the place, with two instances of positive transfer and one of negative transfer.

| World | Transfer Ratio |
|---|---|
| 3x3 | 0.46 |
| 4x4 | 3.06 |
| 5x5 | 1.56 |

Table 1: Transfer Ratio for Varying Domain Sizes

Upon closer inspection, the cause of this instability becomes quite clear. Figures 1 and 2 show the value functions learned for the source and target tasks for the 3x3 and 4x4 grid worlds. Observing these functions, it is clear why the 3x3 grid saw negative transfer: the optimal path in the source task is impossible to traverse in the target task. Conversely, in the 4x4 grid worlds, the optimal path is identical in both the source and target task, despite the fact that the location of the obstacles has completely changed.

These results are enlightening and challenging to my original notions of TL. Whereas I originally hoped to explore the effects of domain size on the practical application of TL, the reality is that small environmental variations seem to overshadow the importance of domain size in practice. This is disappointing because, though the results are obvious upon inspection, manual oversight to predict positive or negative transfer is something we would like to avoid in practice.

Notably, these examples only focus on single pairs of randomly generated source and target tasks, which may be unrepresentative of the general case in the space of each $NxN$ grid world. In order to gauge the more general applicability of TL in a particular domain space, we would need to apply TL to many different pairs in the space.

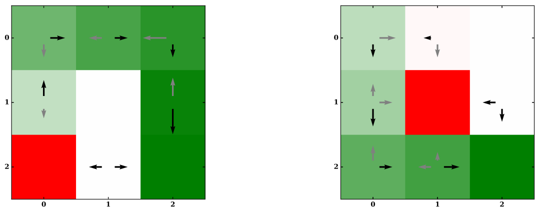

Figure 1: 3x3: Source (Left) and Target (Right) Value Functions

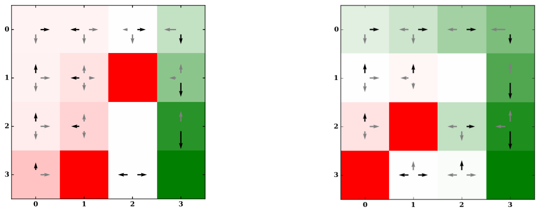

Figure 2: 4x4: Source (Left) and Target (Right) Value Functions

## The 4x4 Worlds

Observing that direct action-value TL is not robust to even small environmental variation under certain circumstances, as well as the computational complexity of testing TL for domains of increasing size, I instead adopt the task of understanding the average performance of TL is for simple navigation tasks. The goal with this set of experiments is to develop a more statistically significant view of the utility of TL. I generate 51 4x4 grid worlds, each with exactly one BLOCKED state and exactly one PIT state. The positions of these, along with the START and GOAL states are randomly chosen. I chose one of the grid worlds to serve as the source task, and apply TL as described in **Methodology** to each of the 50 grid worlds.

The **transfer ratios** for each of the randomly generates worlds are shown in Table 2. I consider any instance of a transfer ratio greater than 1 to be a case of *positive transfer*, and any case with a transfer ratio below 0.99 to be a case of *negative transfer*. All other cases are deemed *neutral*, in which TL appears to have no appreciable impact. Overall, the experiments produced the following breakdown: {Positive: 4, Negative: 11, Neutral 35}.

A few things are interesting to note about these results. First, in the vast majority of cases, transfer had virtually no impact on the asymptotic performance of the agent. This would be fine if TL offered only positive transfer in the non-neutral cases, but we already know this is not the case. In fact, for these experiments I observed negative transfer 3x as often as positive transfer. Closer inspection of cases of positive and negative transfer were much like the results shown in Figure 1 and Figure 2: adversarial variations in obstacles or identical optimal paths are the root causes of negative and positive transfer, respectively.

## Location of Start & Goal

I thought that one of the reasons for such poor performance of TL in the 4x4 space was my introduction of variation in the START and GOAL states. I repeated the experiments with static locations of (0,0) and (2,2) for START and GOAL states, respectively. In this case, I did see more instances of positive and negative transfer (rather than neutral), though the distribution was still skewed towards negative transfer: {Positive: 7, Negative: 23, Neutral: 20}

## Discussion

In this report, I explored direct action-value transfer learning within the Grid World domain. I began by looking at how domain size affects the efficacy of TL, but I quickly abandoned these experiments upon discovering that *small state space variations are much more important than domain size in determining how TL will perform*. I then tried to apply TL to many different environments within the space of 4x4 grid worlds, with both static and random START and GOAL states, with the goal of drawing some general conclusions about what we *generally* expect from random applications of TL. I found that with both static and random START and GOAL states, negative transfer was more common than positive transfer, though in many cases transfer seems to have no impact at all. Though I don't feel comfortable drawing the general conclusion that negative transfer is more common than positive transfer between any two random navigational tasks, these results do suggest that *naive applications of transfer learning may frequently result in no benefits or slowed learning in practice*.

Unfortunately, my biggest take away from these experiments is that simple TL techniques, such as direct copying of action-values, are not only limited in the sense that they can only be applied to highly structurally similar tasks, but are also not at all robust even for tasks with identical or nearly identical state spaces, action spaces, and learning methods. These results ultimately left me disinterested in this flavor of TL and wanting to instead focus on more complex, flexible variants, e.g., those based around options.

## References

Geramifard12, A.; Dann, C.; Klein, R. H.; Dabney, W.; and How, J. P. 2015. Rlpy: A value-function-based reinforcement learning framework for education and research. *Journal of Machine Learning Research* 16:1573–1578.

| Start | Goal | Blocked | Pit | TR |
|---|---|---|---|---|
| (3,1) | (1,2) | (1,1) | (1,3) | 0.79 |
| (0,1) | (3,0) | (0,2) | (1,2) | 0.59 |
| (3,2) | (3,1) | (0,2) | (2,0) | 0.99 |
| (0,2) | (0,3) | (1,3) | (1,1) | 1.33 |
| (1,3) | (1,2) | (2,1) | (2,0) | 0.99 |
| (0,1) | (1,3) | (3,0) | (3,3) | 0.99 |
| (0,1) | (2,1) | (1,1) | (1,3) | 0.99 |
| (2,1) | (3,0) | (3,3) | (2,0) | 0.99 |
| (3,0) | (3,3) | (1,1) | (1,0) | 0.74 |
| (0,1) | (0,0) | (1,0) | (2,0) | 0.99 |
| (0,1) | (3,2) | (2,0) | (3,0) | 0.66 |
| (1,0) | (0,1) | (0,3) | (2,3) | 1.00 |
| (0,0) | (1,0) | (0,3) | (2,3) | 0.99 |
| (2,3) | (2,3) | (3,2) | (2,0) | 0.99 |
| (0,1) | (0,2) | (0,0) | (1,2) | 2.50 |
| (3,0) | (1,2) | (2,0) | (3,3) | 0.39 |
| (2,3) | (1,2) | (3,0) | (2,1) | 1.00 |
| (2,3) | (0,3) | (1,3) | (3,2) | 0.99 |
| (3,2) | (0,0) | (3,0) | (2,0) | 0.99 |
| (0,3) | (1,0) | (2,0) | (2,3) | 0.99 |
| (3,1) | (3,3) | (0,1) | (3,2) | 0.96 |
| (2,0) | (3,2) | (3,1) | (0,0) | 0.74 |
| (3,1) | (1,2) | (2,2) | (0,2) | 0.56 |
| (2,0) | (3,2) | (2,3) | (0,3) | 1.00 |
| (1,1) | (1,3) | (0,2) | (0,0) | 0.99 |
| (0,0) | (3,2) | (3,1) | (2,0) | 0.99 |
| (1,1) | (2,2) | (0,3) | (3,0) | 3.10 |
| (1,0) | (0,3) | (3,0) | (2,0) | 0.99 |
| (3,0) | (2,2) | (0,1) | (0,2) | 0.99 |
| (2,0) | (3,1) | (3,0) | (2,2) | 0.99 |
| (0,3) | (2,2) | (1,3) | (3,1) | 1.01 |
| (0,0) | (0,0) | (1,0) | (0,2) | 0.71 |
| (0,2) | (3,0) | (0,0) | (1,3) | 1.00 |
| (3,1) | (1,2) | (2,1) | (3,0) | 0.99 |
| (0,1) | (3,0) | (3,2) | (1,3) | 0.99 |
| (1,2) | (0,2) | (2,1) | (3,2) | 0.99 |
| (3,2) | (3,3) | (1,2) | (3,0) | 0.35 |
| (2,0) | (0,2) | (3,3) | (0,3) | 0.99 |
| (1,2) | (1,1) | (2,3) | (0,1) | 0.99 |
| (1,2) | (1,0) | (2,0) | (2,3) | 0.99 |
| (0,0) | (1,2) | (1,3) | (2,0) | 0.54 |
| (2,2) | (1,2) | (0,0) | (0,3) | 0.99 |
| (1,1) | (2,3) | (1,0) | (2,0) | 0.99 |
| (3,2) | (1,0) | (0,2) | (1,1) | 0.99 |
| (3,1) | (3,2) | (1,1) | (3,0) | 0.99 |
| (1,3) | (2,0) | (2,3) | (0,3) | 1.00 |
| (1,3) | (0,3) | (2,1) | (3,3) | 0.54 |
| (0,2) | (2,2) | (0,3) | (1,0) | 1.00 |
| (3,3) | (2,1) | (3,1) | (3,2) | 2.13 |
| (2,2) | (3,2) | (2,3) | (2,1) | 0.99 |

Table 2: TL for 4x4 Grid World Variants