ExTra: Transfer-guided Exploration

Anirban Santara∗
Department of Computer Science and Engineering
IIT Kharagpur
Kharagpur, WB 721302
anirban_santara@iitkgp.ac.in

Rishabh Madan∗
Autonomous Ground Vehicle Research Group
IIT Kharagpur
Kharagpur, WB 721302
rishabhmadan96@gmail.com

Balaraman Ravindran
Robert Bosch Centre for Data Science and AI
IIT Madras
Chennai, TN 600036
ravi@cse.iitm.ac.in

Pabitra Mitra
Department of Computer Science and Engineering
IIT Kharagpur
Kharagpur, WB 721302
pabitra@cse.iitkgp.ac.in

Abstract

In this work we present a novel approach for transfer-guided exploration in reinforcement learning that is inspired by the human tendency to leverage experiences from similar encounters in the past while navigating a new task. Given an optimal policy in a related task-environment, we show that its bisimulation distance from the current task-environment gives a lower bound on the optimal advantage of state-action pairs in the current task-environment. Transfer-guided Exploration (ExTra) samples actions from a Softmax distribution over these lower bounds. In this way, actions with potentially higher optimum advantage are sampled more frequently. In our experiments on gridworld environments, we demonstrate that given access to an optimal policy in a related task-environment, ExTra can outperform popular domain-specific exploration strategies viz. epsilon greedy, Model-Based Interval Estimation – Exploration Based (MBIE-EB), Pursuit and Boltzmann in terms of sample complexity and rate of convergence. We further show that ExTra is robust to choices of source task and shows a graceful degradation of performance as the dissimilarity of the source task increases. We also demonstrate that ExTra, when used alongside traditional exploration algorithms, improves their rate of convergence. Thus it is capable of complimenting the efficacy of traditional exploration algorithms.

1 Introduction

While attempting to solve a new task human beings tend to take actions motivated by similar situations faced in the past. These situations could have been encountered for a nearly similar task in an remotely similar environment. But the actions motivated by them oftentimes appear to be good starting points and help in reaching the correct solution faster.

A Reinforcement Learning (RL) agent learns by trial and error in an environment using reward signals that are indicative of progress or accomplishment of the target task [Sutton and Barto [2018]]. The agent uses two policies to act in the environment during the learning phase. The policy that the

∗Authors contributed equally

Preprint. Under review.
agent learns to exploit for solving the target task is known as the target policy. The agent also has a behavioral policy that it uses for exploration in order to find better solutions for the target policy. The sample efficiency and convergence time of an RL algorithm heavily depends on the exploration method used by the behavioral policy. A large body of research in reinforcement learning has been dedicated to the formulation of sample-efficient RL algorithms. Some of the notable developments include count based exploration [Lai and Robbins 1985, Strehl and Littman 2008, Kolter and Ng 2009], curiosity driven exploration [Gregor and Spalek 2014], optimism in the face of uncertainty [Kearns and Singh 2002, Brafman and Tennenholtz 2002, Jaksch et al. 2010], Thompson sampling or posterior sampling and bootstrapped methods [Chapelle and Li 2011, Osband et al. 2013, Agrawal and Jia 2017, Osband et al., 2016], parameter space exploration [Plappert et al. 2017, Fortunato et al. 2017] and intrinsic motivation [Oudeyer and Kaplan 2009, Schmidhuber 2010]. However these methods are based on heuristics specific to the current environment and do not use any prior experiences of the agent in other environments.

The motivation of this work is to formulate an exploration method that uses prior experiences of an agent at solving similar tasks in other environments for improving the efficiency of exploration in the current environment. This problem has previously been studied in the Meta Reinforcement Learning literature. Vezzani et al. [2019] present a representation learning approach that leverages experiences from prior tasks to restrict exploration to the areas of the state space that are relevant to the task at hand. Gupta et al. [2018] propose Model Agnostic Exploration with Structured Noise (MAESN) that introduces structured stochasticity via a latent space that is meta-learned from a set of previously explored tasks. In this work, we take a different approach to solving this problem that is based on the theory of policy-transfer via bisimulation [Castro and Precup 2010]. Given a source environment and the optimal policy for a related task, we calculate its bisimulation distance [Taylor and Stone 2009] from the current task-environment and use it to derive a lower bound on the optimal advantage of state-action pairs of the current environment. During exploration in the current environment, the agent samples actions from a Softmax distribution over these lower bounds. Thus we use transfer learning to guide the exploration in a way that actions with potentially higher advantage are chosen with higher probability. We call this algorithm Transfer-guided Exploration, abbreviated as ExTra. Our experiments on grid-world environments show that ExTra can achieve significant gains in sample complexity and rate of convergence of Q-learning over task-agnostic exploration methods.

Our contributions in this paper can be summarized as follows:

- We prove that transfer via bisimulation [Castro and Precup 2010] maximizes a lower bound on the optimal advantage of target actions.
- We use this bisimulation distance based lower bound to formulate the Transfer-guided Exploration (ExTra) algorithm as a means of using prior experiences for accelerating reinforcement learning in new environments.
- We demonstrate that given the optimal policy from a related task-environment, ExTra achieves faster convergence compared to task-agnostic exploration methods that only use local information, is robust to source task selection with predictable graceful degradation of performance and can compliment traditional exploration methods by improving their rates of convergence.

We briefly describe the essential theoretical concepts used in this paper in Section 2. Then we introduce ExTra in Section 3 and present the results of comparison with traditional exploration methods in Section 4. We conclude the paper with a summary of the proposed method and scope of future work in Section 5.

2 Background

In this section we present a brief introduction to the essential theoretical concepts used in this paper. A Markov Decision Process (MDP) is a discrete time stochastic control process that is commonly used as a framework for reinforcement learning [Sutton and Barto 2018]. An MDP is defined as $M = (S, A, P, r, \rho_0, \gamma)$ where $S$ is the set of states of the environment, $A$ is the set of actions available to the agent, $P: S \times A \times S \rightarrow [0, 1]$ is the transition function that gives a probability
distribution over next states for each state, action pair, \( r : S \rightarrow \mathbb{R} \) is the reward function for the task at hand, \( \rho_0 \) is the initial state distribution and \( \gamma \) is a temporal discount factor. A policy is defined as a function \( \pi : S \rightarrow A \) that returns an action for a given state. The goal of RL is to find the optimal policy \( \pi^* \) that maximizes the cumulated reward received by the agent, \( R(\cdot | r) \).

\[
\pi^* = \arg \max_{\pi} \mathbb{E}[R(\pi | r)]
\]  

(1)

2.1 Exploration in Reinforcement Learning

A reinforcement learning agent learns through trial and error in the environment. At any step of decision making, the agent either "exploits" the best policy it has learned or "explores" other actions in search of a better strategy. Balancing exploration and exploitation is a key challenge in RL and a large body of literature has been dedicated to the formulation of strategies that address this dilemma. Exploration strategies can be widely classified into two categories: directed and undirected [Thrun, 1992]. While directed exploration methods utilize exploration specific knowledge collected online, undirected exploration methods are driven almost purely by randomness with occasional usage of estimates of utility of a state-action pair [Thrun, 1992, Tijsma et al., 2016]. Popular undirected exploration algorithms include random walk [Mozer and Bachrach, 1990], \( \epsilon \)-greedy [Whitehead and Ballard, 1991, Sutton and Barto, 2018] and softmax or Boltzmann exploration [Sutton, 1990, Cesa-Bianchi et al., 2017]. On the other hand, some notable directed exploration methods are count-based [Wiering and Schmidhuber, 1998, Sato et al., 1988], error-based [Schmidhuber, 1991, Thrun and Möller, 1992], and recency based [Wiering and Schmidhuber, 1998]. Next, we describe the exploration algorithms we use as baselines in our experiments. For a detailed review of exploration algorithms in RL, please refer to Tijsma et al. [2016].

2.1.1 \( \epsilon \)-greedy

In \( \epsilon \)-greedy exploration, the agent explores by choosing random actions with probability \( \epsilon \) and follows the learned policy greedily the rest of the time.

2.1.2 Model Based Interval Estimation - Exploration Bonus (MBIE-EB)

MBIE-EB [Strehl and Littman, 2005] is a count-based exploration algorithm that supplies the agent with count-based reward bonuses for favouring exploration of less visited states and actions. The reward bonus is calculated as:

\[
r_{\text{bonus}}(s,a) = \frac{\beta}{\sqrt{n(s,a)}}
\]

(2)

where \( n(s,a) \) is the number of times the agent chose the state action pair \( (s,a) \).

2.1.3 Pursuit

Pursuit [Sutton and Barto, 2018] is an undirected exploration algorithm for Multi-Arm Bandits, adapted for MDP by [Tijsma et al., 2016]. In Pursuit, the agent follows a stochastic policy \( \pi(s, a) \). After the update step, \( t \), if \( a_t^* = \arg \max_a Q_{t+1}(s_t, a) \), Pursuit updates \( \pi_{t+1}(s_t, a) \) as follows:

\[
\pi_{t+1}(s_t, a) = \begin{cases} 
\pi_t(s_t, a) + \beta [1 - \pi_t(s_t, a)], & \text{if } a = a^*_t \\
\pi_t(s_t, a) + \beta [0 - \pi_t(s_t, a)], & \text{if } a \neq a^*_t 
\end{cases}
\]

(3)

where \( \beta \) is a hyperparameter.

2.1.4 Boltzmann exploration

Boltzmann or Softmax exploration [Sutton and Barto, 2018] is an undirected exploration algorithm in which the probabilities of the different actions are assigned by a Boltzmann distribution over the state-action value function \( Q_t(s_t, a) \).

\[
\pi(s_t, a) = \frac{e^{Q_t(s_t, a) / T}}{\sum_{i=1}^{m} e^{Q_t(s_t, a^i) / T}}
\]

(4)
2.2 Transfer in Reinforcement Learning

Transfer learning aims to achieve generalization of skills across tasks by using knowledge gained in one task to accelerate learning of a different task. In reinforcement learning, the transferred knowledge can be high level, such as, rules or advice, sub-task definitions, shaping reward, or low level, such as, task features, experience instances, task models, policies, value functions and distribution priors. Most transfer algorithms for RL make certain assumptions about the relationship between the source and target MDPs. Taylor and Stone [2009] gives a comprehensive overview. In this work, we study the general case of transfer of knowledge between MDPs with discrete state and action spaces and make no assumptions about their structures or relationship. We use the bisimulation transfer framework of Castro and Precup [2010] as it provides a principled method of transfer for the general setting and a way to estimate the relative goodness of actions under the transferred model [Ferns et al., 2004, Taylor et al., 2009].

2.2.1 Transfer using bisimulation metric

Bisimulation, first introduced for MDP by Givan et al. [2003], is a relation that draws equivalence between states of an MDP that have the same long-term behavior. Bisimulation is equivalent to the theory of MDP homomorphism [Ravindran and Barto, 2002; Ravindran, 2003] that studies equivalence relations based on reward structure and transition dynamics. For an MDP, $\mathcal{M} = \langle S, A, P, R, \gamma \rangle$, a relation $E \subseteq S \times S$ is a bisimulation relation if whenever $sEt$, $\forall a \in A, R(s,a) = R(t,a)$ and $\forall a, \forall C \in S/E, \sum_{s',t'} P(s,a)(s') = \sum_{s',t'} P(t,a)(s')$ where $S/E$ is the set of equivalence classes induced by $E$ in $S$. Ferns et al. [2004] proposed bisimulation metric as a quantitative analogue of the bisimulation relation that can be used as a notion of distance between states of an MDP. Under this metric, two states are bisimilar if their bisimulation distance is zero. The higher the distance, the more dissimilar the states are. In order to measure the bisimulation distance between state-action pairs of different MDPs, Taylor et al. [2009] introduced the lax bisimulation metric. Considering two MDPs, $\mathcal{M}_1 = \langle S_1, A_1, P_1, R_1 \rangle$ and $\mathcal{M}_2 = \langle S_2, A_2, P_2, R_2 \rangle$ the lax bisimulation metric is defined as follows.

**Definition 1. Lax bisimulation metric** [Taylor et al., 2009, Castro and Precup, 2010]: Let $M$ be the set of all semi-metrics on $S_1 \times A_1 \times S_2 \times A_2$. $\forall s \in S_1, a \in A_1, t \in S_2, b \in A_2, d \in M$, $F : M \rightarrow M$ is defined as

$$F(d)(s, a, t, b) = c_R |R_1(s, a) − R_2(t, b)| + c_T T_K(d')(P_1(s, a), P_2(t, b))$$

Where $d'(s, t) = \max(\max_{a \in A_1}, \min_{b \in A_2}, d((s, a), (t, b)), \min_{a \in A_1}, \max_{b \in A_2}, d((s, a), (t, b)))$. Then $F$ has a least-fixed point $d_L$ and $d_L$ is called the Lax Bisimulation Metric between $\mathcal{M}_1$ and $\mathcal{M}_2$.

$T_K(d')(P_1(s, a), P_2(t, b))$ is the Kantorovich distance [Gibbs and Su, 2002] between the transition probability distributions. $c_R, c_T \in \mathbb{R}$ are tunable hyperparameters representing relative weightages for the reward and Kantorovich components. Algorithm 1 gives the pseudo-code of the bisimulation based policy transfer algorithm of Castro and Precup [2010]. As only optimal actions of the source domain are the ones that get transferred, the authors define $d_{\approx}(s, (t,b)) = d_L((s, \pi^*(s)), (t, b))$ and use it in their formulations in order to cut down the computation time. Depending upon how $d_{\approx}(s, (t,b))$ and $d'_{\approx}(s, t)$ are related, they define two variants of the lax bisimulation metric – optimistic and pessimistic.

$$d'_{\approx}(s, t) = \begin{cases} \max_{b \in A_2} d_{\approx}(s, (t,b)) & \text{if pessimistic} \\
\min_{b \in A_2} d_{\approx}(s, (t,b)) & \text{if optimistic} \end{cases}$$ (5)
Algorithm 1 Bisimulation based policy transfer [Castro and Precup 2010]

1: for all \( s \in S_1 \), \( t \in S_2 \), \( b \in A_2 \) do
2: \( \text{Compute } d_\infty(s, (t, b)) \)
3: end for
4: for all \( t \in S_2 \) do
5: \( \text{for all } s \in S_1 \) do
6: \( LB(s, t) \leftarrow V_1^*(s) - d_\infty(s, t) \)
7: end for
8: \( s_t \leftarrow \arg\max_{s \in S_1} LB(s, t) \)
9: \( b_t \leftarrow \arg\min_{b \in A_2} d_\infty(s_t, (t, b)) \)
10: end for

Algorithm 2 \( \epsilon \)-greedy Q-learning with Transfer-guided exploration (ExTra)

1: for all \( s \in S_1 \), \( t \in S_2 \), \( b \in A_2 \) do
2: \( \text{Compute } d_\infty(s, (t, b)) \)
3: end for
4: for all \( t \in S_2 \) do
5: \( \text{for all } s \in S_1 \) do
6: \( LB(s, t) \leftarrow V_1^*(s) - d_\infty'(s, t) \)
7: end for
8: \( s_t \leftarrow \arg\max_{s \in S_1} LB(s, t) \)
9: end for
10: step = 0
11: while step < MAXSTEPS do
12: \( \epsilon \)-greedy action \( b \)
13: \( b \leftarrow \pi_{ExTra}(t, M_1, \pi_1^*) \)
14: with probability \( 1 - \epsilon \)
15: \( b \leftarrow \arg\max_{b \in A_2} Q_2(t, b') \)
16: \( r = \text{take\_step}(b) \)
17: \( \text{update}_Q(Q_2(t, b), r) \)
18: step = step + 1
19: end while

3 ExTra: Transfer Guided Exploration

In this section we present Transfer-guided Exploration (ExTra) as a novel directed exploration method for reinforcement learning that is based on the bisimulation transfer framework of [Castro and Precup 2010]. The motivation is to reuse the knowledge acquired in learning an optimal policy in a source domain to accelerate RL in a new (target) domain by improving the efficiency of exploration – especially in the initial stages of learning when the domain-specific statistics used by traditional directed exploration methods are yet to be consistently estimated. We first present some results which relate bisimulation distance to the optimal advantage of an action in a target state.

**Lemma 1.** \( \forall s \in S_1, \forall t \in S_2, \forall b \in A_2, |V_1^*(s) - Q_2^*(t, b)| \leq d_\infty(s, (t, b)) \).

**Proof.**

\[
|V_1^*(s) - Q_2^*(t, b)| = |Q_1^*(s, \pi_1^*(s)) - Q_2^*(t, b)| \\
\leq |R_1(s, \pi_1^*(s)) - R_2(t, b)| + \gamma T_K(d_\infty(P_1(s, \pi_1^*(s)), P_2(t, b))) \\
\text{by similar argument as for Lemma 4 in [Castro and Precup 2010]} \\
= d_\infty(s, (t, b))
\]

**Corollary 1.** \( \forall s \in S_1, \forall t \in S_2, |V_1^*(s) - V_2^*(t)| \leq d_\infty(s, (t, \pi_2^*(t))) \).

**Theorem 1.** Given MDPs, \( M_1 = (S_1, A_1, P_1, R_1) \) and \( M_2 = (S_2, A_2, P_2, R_2) \) and bisimulation metric \( d_\infty : S_1 \times S_2 \times A_2 \rightarrow \mathbb{R} \) we have \( \forall t \in S_2, \forall b \in A_2 \)

\[
A_2^*(t, b) \geq -d_\infty(s_t, (t, b)) - \beta(t),
\]

Where \( A_2^*(t, b) \) is the optimum advantage function in \( M_2, s_t = \arg\max_{s \in S_1} V_1^*(s) - d_\infty'(s, t) \) and \( \beta(t) = d_\infty(s_t, (t, \pi_2^*(t))) \).
We use the optimistic definition of the bisimulation metric in all our experiments as Castro and Precup [2010].

**Proof.** Since, \( V^*_2(t) = \arg \max_{b \in A_2} Q^*_2(t, b), V^*_2(t) - Q^*_2(t, b) \geq 0 \). We have,

\[
V^*_2(t) - Q^*_2(t, b) = |(V^*_2(t) - V^*_1(s_t)) + (V^*_1(s_t) - Q^*_2(t, b))| \\
\leq |V^*_2(t) - V^*_1(s_t)| + |V^*_1(s_t) - Q^*_2(t, b)| \\
\leq d_\infty(s_t, (t, \pi^*_2(t))) + d_\infty(s_t, (t, b)) \quad \text{(from Lemma 1 and Corollary 1)}
\]

\[
\because \ A^*_2(t, b) = Q^*_2(t, b) - V^*_2(t) \geq -d_\infty(s_t, (t, b)) - \beta(t) \quad \square
\]

Theorem 1 gives a lower bound on the optimal advantage of an action in a target state in terms of the bisimulation distance to a source MDP.

**Corollary 2.** The bisimulation transfer algorithm of [Castro and Precup, 2010] maximizes a lower bound on the optimum advantage function \( A^*_2(t, b) \) of the target environment.

**Proof.** In bisimulation transfer, the transferred action \( b_t \) for target state \( t \) is given by \( b_t = \arg \min_{b \in A_2} d_\infty(s_t, (t, b)) = \arg \max_{b \in A_2} -d_\infty(s_t, (t, b)) - \beta(t) \). \quad \square

Note that Theorem 1 and Corollaries 1 and 2 hold for both optimistic and pessimistic versions of lax bisimulation transfer.

**Definition 2.** *Bisimulation Advantage:* Given MDPs, \( M_1 = (S_1, A_1, P_1, R_1) \) and \( M_2 = (S_2, A_2, P_2, R_2) \) and bisimulation metric \( d_\infty : S_1 \times S_2 \times A_2 \rightarrow \mathbb{R} \) we define the bisimulation advantage of an action \( b \in A_2 \) in a state \( t \in S_2 \) as:

\[
A^*_\infty(t, b) = -d_\infty(s_t, (t, b)) - \beta(t)
\]

We use the principle of maximum entropy [Thomas and Cover, 1991] to define a probability distribution on target actions in terms of their bisimulation advantages, \( A^*_\infty(t, b) \) as follows:

\[
\pi^{ExTra}_{\infty}(b|t, M_1, \pi^*_1) = \frac{e^{A^*_\infty(t, b)}}{\sum_{b' \in A_2} e^{A^*_\infty(t, b')}}
\]

In transfer-guided exploration (ExTra), the agent samples actions from \( \pi^{ExTra}_{\infty}(\cdot|t, M_1, \pi^*_1) \). As the optimal policy in the target MDP, \( \pi^*_2 \), is not known during learning, a positive real value can be assigned to \( \beta(t) \) without affecting the overall shape of \( \pi^{ExTra}_{\infty} \) as \( \beta(t) \) is the same for all actions in a state \( t \). We use \( \beta = \alpha n \) in our experiments, where \( \alpha \in \mathbb{R}^+ \) is a tunable hyperparameter and \( n \) is the current step number. As \( n \) grows, \( \pi^{ExTra}_{\infty} \) tends to a uniform distribution over actions. This encourages the agent to explore new actions in the later part of the training. Algorithm 2 gives an example use case of ExTra as an alternative to random exploration in \( \epsilon \)-greedy Q-learning. Please refer to Section C of the Appendix for more ways of using ExTra in conjunction with other traditional exploration methods.

**4 Experimental Results**

In this section we present an empirical analysis of the performance of ExTra. We aim to address the following questions:

1. How does ExTra compare against traditional exploration methods?
2. How sensitive is ExTra to the choice of source task?
3. Can ExTra enhance the performance of other exploration algorithms that only use local information?

We use the optimistic definition of the bisimulation metric in all our experiments as Castro and Precup [2010] note that it gives superior transfer results than the pessimistic definition.
4.1 Evaluation metrics

The goal of this paper is to improve the rate of convergence of RL with the help of external knowledge transferred from a different task. The rate of convergence of an RL algorithm can be judged from a plot of Mean Average Reward (MAR) obtained by the agent over steps of training. Mean Average Reward is defined as follows:

**Definition 3. Mean Average Reward:** The average reward of a trajectory \( \tau \) obtained by following a policy \( \pi \) is the average of all the rewards received by the agent in the trajectory. Mean Average Reward (MAR) of \( \pi \) is the mean of average reward for multiple trajectories rolled out from the policy.

\[
MAR(\pi) = \frac{1}{N} \sum_{i=1}^{N} \frac{1}{T_i} \sum_{t=1}^{T_i} R(s_t, a_t)
\]

Where \( T_i \) is the length of the \( i^{th} \) trajectory.

The Area under the MAR Curve (AuC-MAR) is an objective measure of rate of convergence when the highest MAR values achieved asymptotically by all the candidate algorithms are the same [Taylor and Stone, 2009]. A higher AuC-MAR implies higher rate of convergence. We use AuC-MAR for comparing the rates of convergence of RL algorithms using ExTra with the baseline methods. For the convenience of comparison, we report AuC-MAR as a percentage of AuC for the optimal policy (whose MAR is represented as a straight horizontal line through the steps of training).

For measuring the relative improvement of the rate of convergence achieved by ExTra, we use Transfer Ratio, denoted by \( R \), defined as follows:

\[
R = 100 \times \frac{\text{AuC-MAR with ExTra} - \text{AuC-MAR without ExTra}}{\text{AuC-MAR without ExTra}} \%
\]

4.2 Experimental Design, Results and Discussion

We use stochastic grid-world environments of different levels of complexity (Figures 1 and 3) for analysing the viability of ExTra. All of these environments have the common task of avoiding obstacles and reaching a goal. After being initialised with uniform probability from any of the grid-cells, the agent gets a reward of \(+1\) on reaching the goal and \(0\) elsewhere. The agent has four primitive actions: up, down, left and right. When one of the actions is chosen, the agent moves in the desired direction with \(0.9\) probability, and with \(0.1\) probability it moves uniformly in one of the other three directions or stays in the same place. If the agent bumps into a wall, it does not move and remains in the same state.

We choose Q-learning with four traditional exploration algorithms viz. \( \epsilon \)-greedy uniform random exploration, MBIE-EB, Pursuit and Softmax (described in Section 2) as baselines for comparison. For ExTra, we train the optimal policy for the source domain using Q-learning with \( \epsilon \)-greedy uniform random exploration as it is provably optimal [Sutton and Barto, 2013] and calculate the optimistic bisimulation distance \( d_{\omega}(s, (t, b)), \forall s \in S_1, t \in S_2, b \in A_2 \) tuning \( c_R \) and \( c_T \) for maximum transfer accuracy. We use the PyEMD library by [Mayner, 2018] to calculate earth mover distance for the estimation of \( T_K(d) \) [Pele and Werman, 2008, 2009]. The results reported for each baseline are obtained after rigorous tuning of their respective hyperparameters for the target domain. All MAR and AuC-MAR numbers reported in this paper are averages over 20 different experiments with different random seeds. We tabulate the hyperparameter values in Appendix A and B.

4.2.1 How does ExTra compare against traditional exploration methods?

In our first set of experiments, we show that given an optimal policy for a related task in a related domain, ExTra can obtain superior performance than traditional exploration methods that only use local information. We choose the FourLargeRooms, SixLargeRooms, and NineLargeRooms environments shown in Figure 1 as benchmarks. We train Q-learning agents with \( \epsilon \)-greedy, MBIE-EB, Pursuit and Softmax explorations in these environments as baselines. For ExTra, we choose
Figure 1: Gridworld environments used in our experiments

Figure 2: Variation of MAR with steps of training

FourSmallRooms in Figure 1 as the source environment. We train an \( \epsilon \)-greedy Q-learning agent that uses ExTra instead of uniform random for exploration (Algorithm 2) on each of the three target environments.

Figure 2 shows the variation of MAR over steps of training for the baseline as well as our ExTra agents. Table 1 shows AuC-MAR values. We observe that our ExTra agent consistently achieves faster convergence in all the three environments. This corroborates our claim that ExTra can achieve faster convergence and hence superior sample efficiency if we have access to the optimal policy in a related task-environment.

4.2.2 How sensitive is ExTra to the choice of source task?

In this experiment we study the sensitivity of ExTra to the selection of source task. We choose the SixLargeRooms environment as benchmark. We construct 5 source tasks in this environment where the \( i^{th} \) task has its goal in the location marked with \( i \) in Figure 3. The target task has its goal in the location marked with 6. We report AuC-MAR values obtained by \( \epsilon \)-greedy Q-learning with ExTra for each of the 5 source tasks in Table 2. We also note the best AuC-MAR values obtained by our baseline methods as reference. We make the following observations:

| Target Environment   | \( \epsilon \)-greedy | MBIE-EB | Pursuit | Softmax | ExTra  |
|----------------------|------------------------|---------|---------|---------|--------|
| FourLargeRooms       | 61.08                  | 63.91   | 73.96   | 77.64   | 82.79  |
| SixLargeRooms        | 45.81                  | 54.36   | 63.44   | 61.46   | 72.52  |
| NineLargeRooms       | 43.20                  | 45.87   | 61.52   | 53.66   | 64.81  |

Table 1: Percentage AuC-MAR of \( \epsilon \)-greedy Q-learning with ExTra versus traditional exploration methods.
Table 2: Variation of percentage AuC-MAR of $\epsilon$-greedy Q-learning with ExTra exploration for different choices of source task in SixLargeRooms.

| Baselines       | ExTra from Source # |
|-----------------|---------------------|
| $\epsilon$-greedy | 1 73.7              |
| MBIE-EB         | 2 75.73             |
| Pursuit         | 3 78.54             |
| Softmax         | 4 74.59             |
|                 | 5 81.94             |

Figure 4: Complimentary effect of using ExTra in conjunction with traditional exploration methods

1. Each of our ExTra agents fetch higher AuC-MAR values than any of the baseline methods thus demonstrating the efficacy of ExTra and its robustness over choice of source task.
2. The farther the goal in the source task, the lower is the value of the AuC-MAR. This result is intuitive and demonstrates a graceful degradation in performance of ExTra with increasing dissimilarity of the source task.

4.2.3 Can ExTra enhance the performance of other exploration algorithms that only use local information?

The goal of ExTra is to accelerate convergence of RL through using knowledge transferred from a related task-environment where the optimal policy is known. The goal of this experiment is to show that ExTra can be used alongside a traditional exploration algorithm to improve its rate of convergence. We formulate $\epsilon$-greedy versions of each of our baseline algorithms ($\epsilon = 0.5$), where the agent samples actions from $\pi_{ExTra}$ with probability $\epsilon$ ($\epsilon = 0.5$) and follows the main algorithm rest of the time. Please refer to Appendix C for pseudo-codes of these algorithms. We choose SixLargeRooms as our benchmark environment and FourSmallRooms as source environment for ExTra.

Figure 4 shows MAR plots and Table 3 gives the percentage AuC-MAR scores for each baseline algorithm with and without $\epsilon$-greedy ExTra and the corresponding transfer ratios, $R$. We observe that ExTra can indeed be used as a complimentary exploration method for accelerating the rate of convergence of traditional RL algorithms through the incorporation of transferred knowledge from prior experiences of the agent at related tasks.

5 Conclusion

In this work we present a novel transfer guided exploration algorithm, ExTra, that uses prior knowledge from a related task in a related environment for improving the efficiency of exploration in RL. We demonstrate that our method achieves faster convergence compared to traditional exploration methods that only use local information, is robust to source task selection with predictable graceful degradation of performance and can compliment traditional exploration methods by improving their
Table 3: Comparison of percentage AuC-MAR scores of traditional exploration methods with and without ExTra

|                  | vanilla | with ExTra | \( R \) |
|------------------|---------|------------|---------|
| \( \epsilon \)-greedy Q-learning | 61.91   | 76.10      | 23.41%  |
| MBIE-EB          | 70.55   | 74.87      | 6.11%   |
| Pursuit          | 75.32   | 76.68      | 1.81%   |
| Softmax          | 69.51   | 72.59      | 4.43%   |

rate of convergence. In our future work we plan to extend ExTra to larger state-action spaces and continuous control tasks. An interesting direction of research could be to incorporate ExTra within the meta-RL framework for improving the meta-training and meta-testing sample efficiencies by leveraging knowledge from multiple source MDPs.

Acknowledgements

We gratefully acknowledge Pablo Samuel Castro of Google for helping us with the optimized implementation of bisimulation distance metrics and clarifying some crucial concepts regarding bisimulation based policy transfer in MDPs. Anirban Santara’s work in this project was supported by Google India under the Google India PhD Fellowship Award and Intel under the Intel Student Ambassador Program for Artificial Intelligence.

References

Shipra Agrawal and Randy Jia. Optimistic posterior sampling for reinforcement learning: worst-case regret bounds. In Advances in Neural Information Processing Systems, pages 1184–1194, 2017.

Ronen I Brafman and Moshe Tennenholtz. R-max-a general polynomial time algorithm for near-optimal reinforcement learning. Journal of Machine Learning Research, 3(Oct):213–231, 2002.

Pablo Samuel Castro and Doina Precup. Using bisimulation for policy transfer in mdps. In Twenty-Fourth AAAI Conference on Artificial Intelligence, 2010.

Nicolò Cesa-Bianchi, Claudio Gentile, Gábor Lugosi, and Gergely Neu. Boltzmann exploration done right. In Advances in Neural Information Processing Systems, pages 6284–6293, 2017.

Olivier Chapelle and Lihong Li. An empirical evaluation of thompson sampling. In Advances in neural information processing systems, pages 2249–2257, 2011.

Norm Ferns, Prakash Panangaden, and Doina Precup. Metrics for finite markov decision processes. In Proceedings of the 20th conference on Uncertainty in artificial intelligence, pages 162–169. AUAI Press, 2004.

Meire Fortunato, Mohammad Gheshlaghi Azar, Bilal Piot, Jacob Menick, Ian Osband, Alex Graves, Vlad Mnih, Remi Munos, Demis Hassabis, Olivier Pietquin, et al. Noisy networks for exploration. arXiv preprint arXiv:1706.10295, 2017.

Alison L Gibbs and Francis Edward Su. On choosing and bounding probability metrics. International statistical review, 70(3):419–435, 2002.

Robert Givan, Thomas Dean, and Matthew Greig. Equivalence notions and model minimization in markov decision processes. Artificial Intelligence, 147(1-2):163–223, 2003.

Michal Gregor and Juraj Spalek. Curiosity-driven exploration in reinforcement learning. In 2014 ELEKTRO, pages 435–440. IEEE, 2014.

Abhishek Gupta, Russell Mendonca, YuXuan Liu, Pieter Abbeel, and Sergey Levine. Meta-reinforcement learning of structured exploration strategies. In Advances in Neural Information Processing Systems, pages 5302–5311, 2018.
Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. Journal of Machine Learning Research, 11(Apr):1563–1600, 2010.

Michael Kearns and Satinder Singh. Near-optimal reinforcement learning in polynomial time. Machine learning, 49(2-3):209–232, 2002.

J Zico Kolter and Andrew Y Ng. Near-bayesian exploration in polynomial time. In Proceedings of the 26th Annual International Conference on Machine Learning, pages 513–520. ACM, 2009.

Tze Leung Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. Advances in applied mathematics, 6(1):4–22, 1985.

William Mayner. Fast emd for python: a wrapper for pele and werman’s c++ implementation of the earth mover’s distance metric. https://github.com/wmayner/pyemd, 2018.

Michael C Mozer and Jonathan Bachrach. Discovering the structure of a reactive environment by exploration. In Advances in neural information processing systems, pages 439–446, 1990.

Ian Osband, Daniel Russo, and Benjamin Van Roy. (more) efficient reinforcement learning via posterior sampling. In Advances in Neural Information Processing Systems, pages 3003–3011, 2013.

Ian Osband, Charles Blundell, Alexander Pritzel, and Benjamin Van Roy. Deep exploration via bootstrapped dqn. In Advances in neural information processing systems, pages 4026–4034, 2016.

Pierre-Yves Oudeyer and Frederic Kaplan. What is intrinsic motivation? a typology of computational approaches. Frontiers in neurorobotics, 1:6, 2009.

Ofir Pele and Michael Werman. A linear time histogram metric for improved sift matching. In Computer Vision–ECCV 2008, pages 495–508. Springer, October 2008.

Ofir Pele and Michael Werman. Fast and robust earth mover’s distances. In 2009 IEEE 12th International Conference on Computer Vision, pages 460–467. IEEE, September 2009.

Matthias Plappert, Rein Houthooft, Prafulla Dhariwal, Szymon Sidor, Richard Y Chen, Xi Chen, Tamim Asfour, Pieter Abbeel, and Marcin Andrychowicz. Parameter space noise for exploration. arXiv preprint arXiv:1706.01905, 2017.

Balaraman Ravindran. Smdp homomorphisms: An algebraic approach to abstraction in semi markov decision processes. 2003.

Balaraman Ravindran and Andrew G Barto. Model minimization in hierarchical reinforcement learning. In International Symposium on Abstraction, Reformulation, and Approximation, pages 196–211. Springer, 2002.

Mitsuo Sato, Kenichi Abe, and Hiroshi Takeda. Learning control of finite markov chains with an explicit trade-off between estimation and control. IEEE transactions on systems, man, and cybernetics, 18(5):677–684, 1988.

Jürgen Schmidhuber. Adaptive confidence and adaptive curiosity. In Institut fur Informatik, Technische Universität Munchen, Arcisstr. 21, 800 Munchen 2. Citeseer, 1991.

Jürgen Schmidhuber. Formal theory of creativity, fun, and intrinsic motivation (1990–2010). IEEE Transactions on Autonomous Mental Development, 2(3):230–247, 2010.

Alexander L Strehl and Michael L Littman. A theoretical analysis of model-based interval estimation. In Proceedings of the 22nd international conference on Machine learning, pages 856–863. ACM, 2005.

Alexander L Strehl and Michael L Littman. An analysis of model-based interval estimation for markov decision processes. Journal of Computer and System Sciences, 74(8):1309–1331, 2008.

Richard S Sutton. Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In Machine Learning Proceedings 1990, pages 216–224. Elsevier, 1990.
Richard S Sutton and Andrew G Barto. Reinforcement learning: An introduction. MIT press, 2018.

Jonathan Taylor, Doina Precup, and Prakash Panagaden. Bounding performance loss in approximate mdp homomorphisms. In Advances in Neural Information Processing Systems, pages 1649–1656, 2009.

Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. Journal of Machine Learning Research, 10(Jul):1633–1685, 2009.

Joy A Thomas and TM Cover. Elements of information theory. John Wiley & Sons, Inc., New York. Toni, T., Welch, D., Strelkowa, N., Ipsen, A., and Stumpf, MPH (2009), “Approximate Bayesian computation scheme for parameter inference and model selection in dynamical systems,” Journal of the Royal Society Interface, 6:187–202, 1991.

Sebastian B Thrun. Efficient exploration in reinforcement learning. 1992.

Sebastian B Thrun and Knut Möller. Active exploration in dynamic environments. In Advances in neural information processing systems, pages 531–538, 1992.

Arryoun D Tijsma, Madalina M Drugan, and Marco A Wiering. Comparing exploration strategies for q-learning in random stochastic mazes. In 2016 IEEE Symposium Series on Computational Intelligence (SSCI), pages 1–8. IEEE, 2016.

Giulia Vezzani, Abhishek Gupta, Lorenzo Natale, and Pieter Abbeel. Learning latent state representation for speeding up exploration. arXiv preprint arXiv:1905.12621, 2019.

Steven D Whitehead and Dana H Ballard. Learning to perceive and act by trial and error. Machine Learning, 7(1):45–83, 1991.

Marco Wiering and Jürgen Schmidhuber. Efficient model-based exploration. In Proceedings of the Sixth International Conference on Simulation of Adaptive Behavior: From Animals to Animats, volume 6, pages 223–228, 1998.
Appendix

A  Hyperparameters for Optimistic Bisimulation Transfer

| Transfer Hyperparameters | FourLargeRooms | SixLargeRooms | NineLargeRooms |
|--------------------------|----------------|---------------|---------------|
| $c_R$                    | 0.1            | 0.2           | 0.1           |
| $c_T$                    | 0.9            | 0.9           | 0.9           |
| Threshold                | 0.01           | 0.01          | 0.01          |
| Least fixed point iterations | 5             | 5             | 5             |

B  Hyperparameters for baseline exploration strategies

| Exploration Strategy | $\epsilon$-greedy | Softmax | MBIE-EB | Pursuit | ExTra |
|----------------------|--------------------|---------|---------|---------|-------|
| $\epsilon$-greedy    | Q Learning Rate    | 0.2     | 0.2     | 0.2     | 0.5   |
| $\epsilon$           |                     |         |         |         |       |
| Softmax               | Q Learning Rate    | 0.2     | 0.2     | 0.2     | 0.2   |
| $\tau$               |                     |         | 8.1     |         |       |
| MBIE-EB               | Q Learning Rate    | 0.2     | 0.2     | 0.2     | 0.2   |
| $c_b$                 |                     |         | 0.005   |         |       |
| $\beta$              |                     |         | 0.007   |         |       |
| Pursuit               | Q Learning Rate    | 0.2     | 0.2     | 0.2     | 0.2   |
| $\epsilon$           |                     |         |         |         |       |
| $\beta$              |                     |         |         |         |       |
| ExTra                 | Q Learning Rate    | 0.5     | 0.5     | 0.2     | 1e-6  |
| $\epsilon$           |                     |         |         |         |       |
| $\alpha$             |                     |         |         |         |       |

C  ExTra variants of traditional exploration strategies

Algorithm 3 ExTra + $\epsilon$-greedy Q-learning with random sampling

1:  step = 0
2:  while step < MAXSTEPS do
3:      with probability $\epsilon$
4:         with probability $\epsilon_{bisim}$
5:             $b \sim \pi_{ExTra}(\cdot|t, M_1, \pi_1^*)$
6:         with probability $1 - \epsilon_{bisim}$
7:             $b \sim \text{uniform}(A_2)$
8:      with probability $1 - \epsilon$
9:          $b \leftarrow \text{arg max}_{b'} Q_2(t, b')$
10:  $r = \text{take}_\text{step}(b)$
11:  $update_Q(Q_2(t, b), r)$
12:  step = step + 1
13:  end while
Algorithm 4 ExTra + Softmax
1: step = 0
2: while step < MAXSTEPS do
3:    with probability $\epsilon$
4:        $b \sim \pi_{E_xTra}(\cdot | t, M_1, \pi_1^*)$
5:    with probability $1 - \epsilon$
6:        $b \sim \text{softmax}(Q)$
7:    $r = \text{take}_\text{step}(b)$
8:    $\text{update}_Q(Q_2(t, b), r)$
9:    step = step + 1
10: end while

Algorithm 5 ExTra + Pursuit
1: step = 0
2: $\pi_{\text{pursuit}} = \text{Uniform}(A_2)$
3: while step < MAXSTEPS do
4:    with probability $\epsilon$
5:        $b \sim \pi_{E_xTra}(\cdot | t, M_1, \pi_1^*)$
6:    with probability $1 - \epsilon$
7:        $a \leftarrow \arg\max_{b'} Q_2(t, b')$
8:        $\text{update}_{\pi_{\text{pursuit}}}(a)$
9:        $b \leftarrow \text{Sample}(\pi_{\text{pursuit}})$
10: $r = \text{take}_\text{step}(b)$
11: $\text{update}_Q(Q_2(t, b), r)$
12: step = step + 1
13: end while

Algorithm 6 ExTra + MBIE-EB
1: step = 0
2: $\pi_{\text{pursuit}} = \text{Uniform}(A_2)$
3: while step < MAXSTEPS do
4:    with probability $\epsilon$
5:        $b \sim \pi_{E_xTra}(\cdot | t, M_1, \pi_1^*)$
6:    with probability $1 - \epsilon$
7:        $b \leftarrow \arg\max_{b'} Q_2(t, b')$
8:        $r = \text{take}_\text{step}(b) + \frac{\beta}{\sqrt{n(s, a)}}$
9:    $\text{update}_Q(Q_2(t, b), r)$
10: step = step + 1
11: end while