Proceedings of Machine Learning Research vol 144:1–12, 2021

Learning without Knowing: Unobserved Context in Continuous Transfer Reinforcement Learning

Chenyu Liu CHENYU.LIU018@DUKE.EDU
Yan Zhang YAN.ZHANG2@DUKE.EDU
Yi Shen YI.SHEN478@DUKE.EDU
Michael M. Zavlanos MICHAEL.ZAVLANOS@DUKE.EDU
Department of Mechanical Engineering and Material Science
Duke University, Durham, NC 27708, USA

Abstract

In this paper, we consider a transfer Reinforcement Learning (RL) problem in continuous state and action spaces, under unobserved contextual information. For example, the context can represent the mental view of the world that an expert agent has formed through past interactions with this world. We assume that this context is not accessible to a learner agent who can only observe the expert data. Then, our goal is to use the context-aware expert data to learn an optimal context-unaware policy for the learner using only a few new data samples. Such problems are typically solved using imitation learning that assumes that both the expert and learner agents have access to the same information. However, if the learner does not know the expert context, using the expert data alone will result in a biased learner policy and will require many new data samples to improve. To address this challenge, in this paper, we formulate the learning problem as a causal bound-constrained Multi-Armed-Bandit (MAB) problem. The arms of this MAB correspond to a set of basis policy functions that can be initialized in an unsupervised way using the expert data and represent the different expert behaviors affected by the unobserved context. On the other hand, the MAB constraints correspond to causal bounds on the accumulated rewards of these basis policy functions that we also compute from the expert data. The solution to this MAB allows the learner agent to select the best basis policy and improve it online. And the use of causal bounds reduces the exploration variance and, therefore, improves the learning rate. We provide numerical experiments on an autonomous driving example that show that our proposed transfer RL method improves the learner's policy faster compared to existing imitation learning methods and enjoys much lower variance during training.

Keywords: Transfer reinforcement learning, causal inference, multi-armed bandit algorithms.

1. Introduction

Reinforcement Learning (RL) has been widely used recently to solve stochastic multi-stage decision making problems in high dimensional spaces with unknown models, in areas as diverse as robotics Zhu et al. (2017) and medicine Raghu et al. (2017). Various methods have been proposed for this purpose that generally require large amounts of data to learn good control policies. Among these methods, Transfer Learning (TL) aims at addressing data efficiency in RL by using data from expert agents to learn optimal policies for learner agents with only a few new data samples Taylor and Stone (2009); Lazaric (2012). Nevertheless, a common assumption in TL methods is that both the expert and learner agents have access to the same information. This is not always the case in practice. For example, human drivers often develop a mental view of the world through past experiences that

along with other sensory information, e.g., visual information, guides their actions. This additional experiential information cannot be obtained using, e.g., camera and Lidar sensors found onboard autonomous vehicles. As a result, using expert data without knowing the expert context can generate learned policies that are biased and may require many new data samples to improve.

In this paper, we propose a new transfer RL method in continuous state and action spaces that can deal with unobservable contextual information. Specifically, we model the expert and learner agents using contextual Markov Decision Processes (MDP) Hallak et al. (2015), where the state transitions and the rewards are affected by contextual variables. Moreover, we assume that the expert agent can observe this contextual information and can generate data using a context-aware policy. The goal is to use the experience from the expert agent to enable a learner agent, who cannot observe the contextual information, to learn an optimal context-unaware policy with much fewer new data samples. In related literature, the contextual variables in contextual MDPs are typically used to model different tasks that the agents can accomplish. In these problems, transfer RL enables the use of expert data from many different source tasks to accelerate learning for a new desired target task, see, e.g., Taylor et al. (2007); Lazaric et al. (2008); Konidaris et al. (2012); Barreto et al. (2017); Tirinzoni et al. (2019). These methods typically assume that some or all source tasks share the same reward or transition functions with the target task. As a result, the models learned from the expert's data can be easily transfered to the learner agent. However, as discussed in Zhang and Bareinboim (2017), when contextual information about the world is unobserved, the constructed reward or transition models can be biased, and this can introduce errors in the policies learned by the learner agent.

Related is also recent work on Learning from Demonstrations (LfD); see, e.g., Ho and Ermon (2016); Hausman et al. (2017); Li et al. (2017); Torabi et al. (2018); Lee et al. (2019); Ghasemipour et al. (2020). Specifically, Ho and Ermon (2016); Torabi et al. (2018) propose a method to learn a policy maximizing the likelihood of generating the expert's data, while Ghasemipour et al. (2020) propose a method to learn a policy that matches the expert-specified state distribution. However, these works assume that the expert and learner agents observe the same type of information. When the expert makes decisions based on additional contextual variables that are unknown to the learner, the imitated policy function can be suboptimal, as we later discuss in Section 2. On the other hand, Hausman et al. (2017); Li et al. (2017); Lee et al. (2019) propose a method to learn a policy parameterized by latent contextual information included in the expert's dataset. Therefore, given a contextual variable, the learner agent can directly execute this policy. However, if the context can not be observed, these methods cannot be used to learn a context-unaware policy. LfD with contextual information is also considered in de Haan et al. (2019); Etesami and Geiger (2020); Zhang et al. (2020). Specifically, de Haan et al. (2019); Etesami and Geiger (2020) assume that the expert's observations are accessible to the learner so that the environment model can be identified. On the other hand, Zhang et al. (2020) provide conditions on the underlying causal graphs that guarantee the existence of a learner policy matching the expert's performance. However, such a policy does not necessarily exist for general TL problems where unobservable contextual information may affect the expert agent's decisions. Along with LfD, related is also literature on one/few-shot reinforcement learning Duan et al. (2017); Yu et al. (2019); Seyed Ghasemipour et al. (2019), where the objective is to retrain an trained policy for a new unseen scenario using only one or a few new expert demonstrations for this new scenario. However, in the TL problem considered here, no expert demonstrations are available for the unseen testing cases in which the contextual information is hidden and sampling of the contextual variable cannot be controlled.

To the best of our knowledge, most closely related to the problem studied in this paper are the works in Zhang and Bareinboim (2017); Zhang and Zavlanos (2020). Specifically, Zhang and Bareinboim (2017) propose a TL method for Multi-Armed-Bandit (MAB) problems with unobservable contextual information using the causal bounds. Then, Zhang and Zavlanos (2020) extend this framework to general RL problems by computing the causal bounds on the value functions. However, the approach in Zhang and Zavlanos (2020) can only handle discrete state and action spaces and does not scale to high dimensional RL problems. In this paper, we propose a novel TL framework under unobservable contextual information which can be used to solve continuous RL problems. Specifically, we first learn a set of basis policy functions that represent different expert behaviors affected by the unobserved context, using the unsupervised clustering algorithms in Hausman et al. (2017); Li et al. (2017); Lee et al. (2019). Although the return of each data sample is included in the expert dataset, as discussed in Zhang and Bareinboim (2017), it is impossible to identify the expected returns of the basis policy functions when the context is unobservable. Therefore, the basis policy functions cannot be transferred directly to the learner agent. Instead, we use them as arms in a Multi-Armed Bandit (MAB) problem that the learner agent can solve to select the best basis policy function and improve it online. Specifically, we propose an Upper Confidence Bound (UCB) algorithm to solve this MAB that also utilizes causal bounds on the expected returns of the basis policy functions computed using the expert data. These causal bounds allow to reduce the exploration variance of the UCB algorithm and, therefore, accelerate its learning rate, as shown also in Zhang and Bareinboim (2017). We provide numerical experiments on an autonomous driving example that show that our proposed TL framework requires much fewer new data samples than existing TL methods that do not consider the presence of unobserved contextual information and learn a single policy function using the expert dataset.

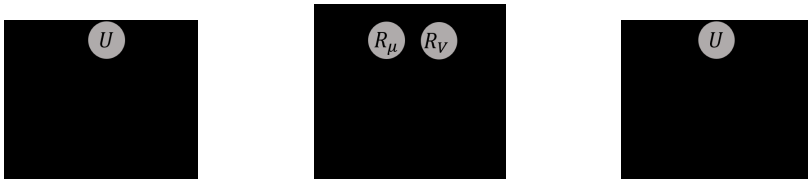
The rest of this paper is organized as follows. In Section 2, we define the proposed TL problem in continuous state and action spaces with unobservable contextual information. In Section 3, we formulate the MAB problem for the learner agent and develop the proposed causal bound-constrained UCB algorithm to solve it. In Section 4, we provide numerical experiments on an autonomous driving example that illustrate the effectiveness of our method. Finally, in Section 5, we conclude our work.

2. Preliminaries and Problem Definition

In this section, we define the contextual Markov Decision Processes (MDP) used to model the RL agents, and formulate the TL problem that uses experience data from a context-aware expert agent to learn an optimal policy for a context-unaware learner agent.

2.1. Contextual MDPs

Consider a contextual MDP defined by the tuple $\left(s_t, a_t, f^U(s_{t+1}|s_t, a_t), r^U(s_t, a_t), \rho_0\right)$ as in Hallak et al. (2015), where $s_t \in \mathcal{S}$ represents the state of the agent at time t, and $a_t \in \mathcal{A}$ denotes the action of the agent at time t. Unlike Zhang and Zavlanos (2020), in this paper, we assume that the state and action spaces $\mathcal{S}$ and $\mathcal{A}$ can be either discrete or continuous. The transition function $f^U(s_{t+1}|s_t, a_t)$ represents the probability of transitioning to state s_{t+1} after applying action a_t at state s_t, the reward function $r^U(s_t, a_t)$ denotes the reward received by the agent after applying action a_t at state s_t, and $\rho_0 : \mathcal{S} \to \mathbb{R}$ represents the distribution of the initial state s_0. In a contextual MDP, the transition and reward functions are parameterized by a context variable $U \in \mathcal{U}$, where $\mathcal{U}$ is a discrete or continuous space. The random variable U is sampled from a stationary distribution $P(U)$ at the



(a) Expert (b) Auxiliary expert (c) Learner

Figure 1: Expert and learner causal graphs. (a) represents the causal graph of expert; (b) represents the auxiliary causal graph of expert where the context U is replaced by $\{R_\mu, R_V\}$; (c) denotes the causal graph of the learner. The gray nodes represent the unobservable context U, and the white nodes represent observable data. The arrows indicate the directions of the causal effects.

beginning of each episode. The transition and reward functions are parameterized by contextual variables in many practical control and learning problems. For example, in human-in-the-loop control Zhou et al. (2020), the context U can represent hidden human preference and affect the reward function. In what follows, we define the context-aware policy function of the expert by $\pi_e(s_t, a_t, U) : \mathcal{S} \times \mathcal{A} \times \mathcal{U} \to [0, 1]$, that represents the probability of selecting action a_t at state s_t given the contextual variable U. We also assume that the expert implements a context-aware policy π_e^* that approximately solves the problem $\max_{\pi_d} \mathbb{E}_{\rho_0}\big[\sum_{t=0}^{T} \gamma^t r^U(s_t, a_t)|\pi_e(s_t, a_t, U)\big]$ for all $U \in \mathcal{U}$, where $\gamma \leq 1$ is the discount factor. Note that the expert policy π_e^* does not need to solve the RL problem exactly. This way we can model, e.g., human experts that occasionally make mistakes. Finally, we assume that the learner agent cannot observe the contextual information U. Therefore, the learner implements a context-unaware policy function $\pi(s_t, a_t) : \mathcal{S} \times \mathcal{A} \to [0, 1]$ that denotes the probability of selecting action a_t only based on state s_t. The goal of the learner agent is to find an optimal policy $\pi^*(s_t, a_t)$ which solves the problem

$$\max_{\pi} \mathbb{E}_{\rho_0, P(U)}\big[\sum_{t=0}^{T} \gamma^t r^U(s_t, a_t)|\pi(s_t, a_t)\big]. \tag{1}$$

2.2. Experience Transfer under Unobserved Contextual Information

Let $\tau_D = \{\tau_1, \tau_2, \ldots, \tau_N\}$ denote a dataset generated by the expert, where $\tau_i = \{s_0^i, a_0^i, s_1^i, a_1^i, \ldots, s_T^i, a_T^i, V^i\}$ represents the state-action trajectory collected during the ith execution of the context-aware policy π_e^*, and $V^i = \sum_{t=0}^{T} \gamma^t r^U(s_t^i, a_t^i)$ is the total reward received at the end of this episode. Specifically, to generate the expert data, at the beginning of episode i, we sample the context variable U^i independently from the distribution $P(U)$. Then, given the sampled context U^i, the expert interacts with the environment using policy function $\pi_e^*(s_t^i, a_t^i, U^i)$ and collects data τ_i. The contextual information U^i is not recorded in dataset τ_D. The expert data collection process is reflected in the causal graph in Figure 1(a). Finally, the dataset τ_D without the contextual information U is transferred to a learner agent with the goal to learn a context-unaware policy π, as described in the causal graph in FIgure 1(c). Specifically, we aim to solve the following problem.

Problem 1 *Given the experience data τ_D generated by an expert agent that uses a context-aware policy π_e^*, design a transfer learning algorithm for a learner agent that uses the data τ_D excluding the contextual information to learn the optimal context-unaware policy $\pi^*(s_t, a_t)$ that solves the optimization problem (1) with only a few new data samples.*

(a)

$r^U(a_t)$	$U=0$	$U=1$
$a_t=0$	1	11
$a_t=1$	3	10

(b)

$P(a_t, r^U)$	$r^U=1$	$r^U=3$	$r^U=10$	$r^U=11$
$a_t=0$	0.05	0	0	0.45
$a_t=1$	0	0.45	0.05	0

Table 1: Reward and Observational Distribution for the 2-arm Bandit Problem in Example 1

In the absence of any unobserved contextual effects, Problem 1 has been studied in Ho and Ermon (2016); Torabi et al. (2018). In these works, a single policy function that generates the experience dataset τ_D with the maximum likelihood is computed and is directly transferred to the learner agent. When the dataset τ_D is generated as per the causal graph in Figure 1(a), the methods in Ho and Ermon (2016); Torabi et al. (2018) essentially compute a mixture of the context-aware policy π_e^* based on the context distribution $P(U)$. On the other hand, in the presence of unobserved contextual information, a possible approach to Problem 1 is to first use the unsupervised clustering methods in Hausman et al. (2017); Li et al. (2017) on the dataset τ_D to obtain a cluster of basis policy functions $\{\mu_k\}_{k=1,2,\dots,K}$, so that each function $\mu_k(s_t, a_t) : \mathcal{S} \times \mathcal{A} \to [0, 1]$ approximates the underlying true expert behavior π_e^* associated with a given context. Then, we can select the basis policy function with the best empirical expected return in the dataset τ_D. Next, we give an example to show that both these approaches can lead to a policy function that is far from the true optimal policy.

Example 1 *Consider a 2-arm bandit problem, where the action is binary, i.e., $a_t \in \{0, 1\}$. At the beginning of each episode, a binary contextual variable $U \in \{0, 1\}$ is sampled from a Bernoulli distribution $P(U = 0) = \frac{1}{2}$. The binary variable U affects the reward $r^U(a_t)$ received during each episode according to Table 1 (a). The expert agent can observe the contextual variable U and executes a context-aware policy $\pi_e^*(a_t, U)$, such that $\pi_e^*(a_t = 1, U = 0) = 0.9$ and $\pi_e^*(a_t = 0, U = 1) = 0.9$. As a result, the expert generates multiple data points $\{a_t, r_t\}$ to obtain an observational distribution $P(a_t, r_t)$ as shown in Table 1 (b). Denote the true expected reward when taking action $a_t = 0$ as $\mathbb{E}\big[r_t | do(a_t = 0)\big]$. Then, we have that $\mathbb{E}\big[r_t | do(a_t = 0)\big] = 6$ and $\mathbb{E}\big[r_t | do(a_t = 1)\big] = 6.5$. The true optimal policy is to select $a_t = 1$ with probability 1.*

Given the underlying contextual variable distribution $P(U = 0) = 1/2$, the Direct Imitation strategy in Ho and Ermon (2016); Torabi et al. (2018) learns a policy $\pi_{DI}(a_t = 0) = 1/2$ and $\pi_{DI}(a_t = 1) = 1/2$, which is far from the true optimal policy. On the other hand, in this 2-arm bandit problem, the basis policy function can be directly obtained as $\{\mu_1, \mu_2\}$, where $\mu_1(a_t = 0) = 1$ and $\mu_2(a_t = 1) = 1$. The empirical estimates of the expected rewards for these two basis policies can be computed using the observational distribution in Table 1 (b). Specifically, we have that $\mathbb{E}[r^U | a_t = 0] = 10$ and $\mathbb{E}[r^U | a_t = 1] = 3.7$. This suggests that we should transfer the basis policy function μ_1 to the learner agent, which is the opposite of the true optimal policy.

Example 1 shows that selecting the mixture policy π_{DI} or the basis policy function with the best observed rewards can both lead to sub-optimal policies. In addition, unlike the bandit problem in Example 1, for general continuous RL problems, the optimal context-unaware policy π^* may not belong to the set of basis policy functions obtained using the algorithms in Hausman et al. (2017); Li et al. (2017). In the next section, we propose a new TL framework that can address these challenges.

3. Continuous Transfer Learning with Unobserved Contextual Information

In this section, we propose a TL framework to solve Problem 1. First, we compute a set of basis policies from the dataset τ_D. Then, we formulate a MAB problem that uses these basis policies as

Algorithm 1 Transfer RL under Unobserved Contextual Information

Input: The set of basis policy functions $\{\mu_k\}$, the causal bounds on the expected total rewards of each policy function $[l_k, h_k]$, a non-increasing function $f(i)$ and $T_k(0) = 1$ for all k.

1: Remove any arm k with $h_k < l_{max}$, where $l_{max} = \max_k\{l_k\}$.
2: Execute all basis policies μ_k once and record the resulting accumulated rewards $\{\hat{V}_k(0)\}$.
3: **for** episode $i = 1, 2, \ldots$ **do**
4: For every basis policy function μ_k, compute $\hat{H}_k(i) = \min\{H_k(i), h_k\}$, where

$$H_k(i) = \hat{V}_k(i-1) + \sqrt{\frac{2f(i)}{T_k(i-1)}}; \tag{2}$$

5: Select the basis policy function μ_{k^*} such that $k^* = \arg\max_k \hat{H}_k(i)$;
6: Execute the policy μ_{k^*} during episode i and update it using policy optimization algorithms, e.g., PPO;
7: At the end of the episode, compute the accumulated rewards $V(i)$ and update $\hat{V}_{k^*}(i)$ as $\hat{V}_k(i) = \frac{1}{i+1}V(i) + \frac{i}{i+1}\hat{V}_k(i-1)$ if $k = k^*$, and $\hat{V}_k(i) = \hat{V}_k(i-1)$ otherwise;
8: Update the number of times every basis policy function has been selected as $T_k(i) = T_k(i-1) + 1$ if $k = k^*$, and $T_k(i) = T_k(i-1)$ otherwise.
9: **end for**
10: Output policy μ_{k^*}.

its arms and the learner agent can learn to select the best basis policy function to improve online. Finally, to reduce the exploration variance of the MAB problem, we constrain exploration using causal bounds on the accumulated rewards that can be achieved by these basis policy functions.

3.1. Multi-Armed Bandit Formulation

The first step in defining a MAB model for Problem 1 is to use the unsupervised clustering methods in Hausman et al. (2017); Li et al. (2017) to obtain a cluster of basis policy functions $\{\mu_k\}_{k=1,2,\ldots,K}$ from the expert dataset τ_D, as discussed in Section 2.2. Each function $\mu_k(s_t, a_t) : \mathcal{S} \times \mathcal{A} \to [0, 1]$ represents a distinct expert behavior under a specific context, e.g., driving cautiously or aggressively. We assume that the expert policy $\pi_e^*(s_t, a_t, U)$ is a mixture of the basis policies $\{\mu_k\}_{k=1,2,\ldots,K}$, i.e., $\pi_e^*(s_t, a_t, U) = \sum_{k=1}^{K} \mu_k(s_t, a_t)P(\mu_k|U)$, where $P(\mu_k|U)$ denotes the conditional probability of selecting the basis policy μ_k when context U is observed. Through this policy transformation, we obtain an auxiliary causal graph for the expert, as described in Figure 1(b). And the dataset τ_D can be written as $\{m_i, V_i\}$, where $m_i \in \{1, 2, \ldots, K\}$ represents the index of the basis poilcy function that generates the ith trajectory τ_i in dataset τ_D and V_i denotes its accumulated rewards.

Next, we use the set $\{\mu_k\}$ as initial policy parameters. Each one of these policy parameters gives rise to an arm in a MAB problem, which we solve using the Upper Confidence Bound (UCB) algorithm to obtain the optimal initial policy in $\{\mu_k\}$. Then, we can improve this policy through online interactions using any existing policy optimization algorithm, e.g., the Proximal Policy Optimization (PPO) algorithm in Schulman et al. (2017) with experience replay. This process is presented in Algorithm 1. Specifically, we start by evaluating the basis policy functions μ_k by executing them in the environment once and computing their accumulated rewards $\{\hat{V}_k(0)\}$ (line 2). Then, using these estimates of the total rewards we can compute the UCBs on the expected values of the basis policy functions $H_k(i)$ according to (2) (line 4). Next, we select the basis policy function μ_{k^*} that has the highest UCB on its expected value (line 5), execute this policy for the duration of

one episode, and improve it using the PPO (line 6). At the end of each episode, we compute the total rewards of the improved basis policy μ_{k^*}, and update the estimate of the expected value $\hat{V}_{k^*}$ using this policy (line 7) as well as the number of times T_k this policy has been selected (line 8). Finally, Algorithm 1 returns to line 4 and a new episode is initialized.

Note that the UCB algorithm proposed in Algorithm 1 gradually explores more frequently the most promising basis policy function that performs well initially and is improved at the fastest rate using new online data. However, this UCB algorithm may suffer large variance during exploration, which affects its learning rate. In the next section, we provide bounds on the expected returns of each basis policy function $\mathbb{E}\big[V|do(\mu_k)\big]$ using the expert data τ_D and transfer these bounds to the UCB algorithm to reduce its exploration variance.

3.2. Causal Bound Constrained MAB

When the contextual variable is missing from the experience data τ_D, as discussed in Zhang and Bareinboim (2017), the true expected reward $\mathbb{E}[V|do(\mu)]$ cannot be identified. However, it is possible to obtain bounds on $\mathbb{E}[V|do(\mu)]$ using the observational data $\{m_i, V_i\}$, similar as Zhang and Bareinboim (2020). To do so, define the unobserved contextual variable U by a pair of random variables (R_μ, R_V), where R_μ has support $\{1, 2, \dots, K\}$ and R_V has support $\mathbb{R}$. The random variable R_μ represents the basis policy function that the expert decides to adopt in practice, i.e., $\mu_{R_\mu} = f_\mu(R_\mu)$, while the random variable R_V represents the accumulated rewards received when the expert executes policy μ_k, i.e., $R_V = f_V(\mu_k, R_V)$. The causal relationship betweeen the observed expert policy μ, the observed accumulated rewards V, and the unobserved random variable pair (R_μ, R_V) is shown in Figure 1(b). If the RL problem has bounded rewards and is episodic, then the random accumulated rewards R_V have a compact support $R_V \in [V_l, V_u]$. Then, given the observed distribution $P(\mu, V)$, we can obtain the following bounds on the causal effect $\mathbb{E}[V|do(\mu)]$.

Theorem 3.1 *If $R_V \in [V_l, V_u]$, then we have that*

$$\mathbb{E}\big[V|\mu_k\big]P(\mu_k) + (1 - P(\mu_k))V_l \leq \mathbb{E}\big[V|do(\mu_k)\big] \leq \mathbb{E}\big[V|\mu_k\big]P(\mu_k) + (1 - P(\mu_k))V_u. \quad (3)$$

Furthermore, let $\mathbb{E}\big[V|\mu_j, R_\mu = k\big]$ represent the expected return when the expert agent counterfactually executes policy μ_j instead of the policy μ_k that was used by the expert to generate the data, so that $R_\mu = k$. If $\mathbb{E}\big[V|\mu_j, R_\mu = k\big] \leq \mathbb{E}\big[V|\mu_k, R_\mu = k\big]$ for all $j \neq k$, then we have that

$$\mathbb{E}\big[V|\mu_k\big]P(\mu_k) + (1 - P(\mu_k))V_l \leq \mathbb{E}\big[V|do(\mu_k)\big] \leq \sum_{j=1}^{K} \mathbb{E}\big[V|\mu_j\big]P(\mu_j). \quad (4)$$

Proof Recall that

$$\mathbb{E}[V|do(\mu_k)] = \int_{R_\mu = k, R_V} f_V(\mu_k, R_V) dP(R_\mu = k, R_V) + \int_{R_\mu \neq k, R_V} f_V(\mu_k, R_V) dP(R_\mu \neq k, R_V), \quad (5)$$

where the first term in (5) represents the expected accumulated rewards of the trajectories observed in the dataset τ_D that are generated using policy μ_k, i.e.,

$$\begin{aligned}\int_{R_\mu = k, R_V} f_V(\mu_k, R_V) dP(R_\mu = k, R_V) &= P(R_\mu = k) \int_{R_V} f_V(\mu_k, R_V) dP(R_V | R_\mu = k) \\ &= \mathbb{E}\big[V|\mu_k\big]P(\mu_k). \quad (6)\end{aligned}$$

On the other hand, the second term in (5) represents the counterfactual accumulated rewards received when executing policy μ_k, although other policy functions $\mu_{j\neq k}$ are actually executed in the dataset τ_D. Since policy μ_k was not actually executed when $R_\mu \neq k$, its counterfactual accumulated rewards are never observed in expert dataset τ_D, and we can only bound $\int_{R_\mu\neq k,R_V} f_V(\mu_k,R_V)dP(R_\mu \neq k,R_V)$ using the bounds on the random variable R_V, i.e.,

$$V_l\big(1-P(\mu_k)\big) \leq \int_{R_\mu\neq k,R_V} f_V(\mu_k,R_V)dP(R_\mu \neq k,R_V) \leq V_u\big(1-P(\mu_k)\big). \tag{7}$$

Substituting the bounds (6) and (7) into (5), we obtain the bound in (3).

Now, assuming that $\mathbb{E}\big[V|\mu_j,R_\mu=k\big] \leq \mathbb{E}\big[V|\mu_k,R_\mu=k\big]$ for all $j\neq k$, we have that $\int_{R_\mu=k,R_V} f_V(\mu_j,R_V)dP(R_\mu=k,R_V) \leq \int_{R_\mu=k,R_V} f_V(\mu_k,R_V)dP(R_\mu=k,R_V)$ for every $j\neq k$. Using this bound, we obtain that

$$\int_{R_\mu\neq k,R_V} f_V(\mu_k,R_V)dP(R_\mu \neq k,R_V) \leq \sum_{j\neq k} f_V(\mu_j,R_V)dP(R_\mu=j,R_V). \tag{8}$$

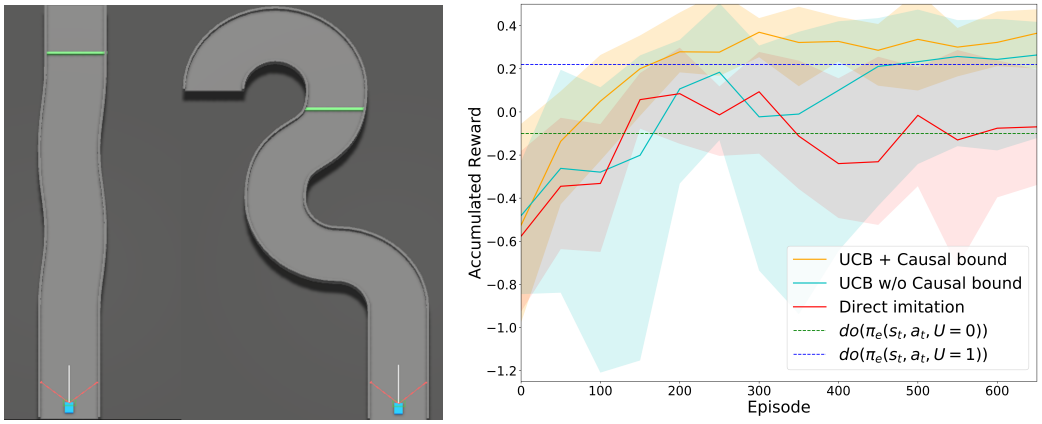
Combining the bound (8) with (5), we obtain the bound in (4). The proof is complete. ■

Note that the assumption $\mathbb{E}\big[V|\mu_j,R_\mu=k\big] \leq \mathbb{E}\big[V|\mu_k,R_\mu=k\big]$ in Theorem 3.1 suggests that the basis policy μ_k implemented by the expert is better than any other policy in expectation. This holds when the expert policy is close to the optimal context-aware poilcy $\pi_e^*(a_t,s_t,U)$. Same as in Zhang and Bareinboim (2017), here too we can incorporate in Algorithm 1 the bounds $[l_k,h_k]$ on the causal effect $\mathbb{E}\big[V|do(\mu_k)\big]$ computed using Theorem 3.1 for every basis policy function μ_k. To do so, we compare the bound H_k in (2) computed using online data to the upper bound h_k, and select the tighter one as an estimate for the UCB $\hat{H}_k$ (line 4 in Algorithm 1). Observe that during the early learning stages, the UCB H_k computed using new online data can be over-optimistic for suboptimal basis policies. The causal bound h_k obtained using the expert's experience can tighten these bounds and, therefore, reduce the number of queries to suboptimal basis policies. This can significantly improve the efficiency of the data collection for the learner by reducing the variance of the UCB exploration, as we also show in Section 4.

4. Numerical Experiments

In this section, we demonstrate the effectiveness of the proposed transfer RL algorithm on an autonomous racing car example. Specifically, the goal is to transfer driving data from an expert driver that has a global mental view of the race track (e.g., from past experience driving on this track) to a learner autopilot that can only perceive the race track locally using the vehicle's on-board sensors. In practice, the expert's decisions will be affected by their mental view of the race track, e.g., the anticipated shape of the track ahead. However, this information is not recorded in the expert's data and, therefore, is not available to the learner autopilot. As such, it constitutes unobservable context.

We simulate the above problem using the Unity Physics Engine, Juliani et al. (2018). Specifically, we consider the two types of tracks shown in Figure 2(*a*)subfigure. Let $U=0$ denote the straight track and $U=1$ denote the curved track. We assume that the contextual variable U is sampled from a Bernoulli distribution $P(U=0)=0.7$ at the beginning of each episode. The green lines in both tracks represent the finish line. In Figure 2(*a*)subfigure, the vehicle (blue box) is equipped with an



(*a*) (*b*)

Figure 2: In (a), two types of tracks are presented. The straight track denotes the environment under $U = 0$ and the curved track denotes the environment under $U = 1$; In (b), the performance of Algorithm 1 (orange), Algorithm 1 without causal bounds (blue) and transferring a single policy learnt by direct imitation learning (red) are presented. Each curve is obtained by running 10 trials.

onboard sensor (three lines emitting from the car) that detects the track boundaries, if within sensor range, and their distance to the vehicle. The vehicle takes actions in the set $\mathcal{A} := \{0, 1, 2, 3, 4\}$, where action 0 means "maintain current speed", action 1 means "accelerate", action 2 means "decelerate", action 3 means "turn clockwise", and action 4 means "turn counterclockwise". The policy of the learner autopilot is denoted by $\pi(s_t, a_t)$, where the state s_t represents the vehicle's orientation, velocity, last-step action, and the sensor measurements. On the other hand, the policy of the expert driver is denoted by $\pi_e^*(s_t, a_t, U)$, where the context variable U denotes the type of the track that the expert has experience driving on. The goal is to drive the vehicle to the finish line in the shortest period of time while avoiding collisions with the track boundaries. To achieve this, let the learner autopilot receive a reward $r = 1$ when it reaches the finish line and $r = -0.08$ when the vehicle collides with the track boundaries. Otherwise, the agent receives a reward $r = -0.003$. Each episode ends when the vehicle reaches the finish line or when 500 time steps have elapsed. We trained the expert policies $\pi_e^*(s_t, a_t, U = 0 \text{ or } 1)$ for each track independently using the PPO algorithm. Then, we executed the expert policies 300 times to construct the dataset τ_D as discussed in Section 2.2.

After constructing the dataset τ_D, we transfer it to the learner autopilot and cluster the trajectories in τ_D into two sets using the K-means algorithm, where the number of clusters can be determined using Principle Component Analsysis. Then, using behavior cloning Torabi et al. (2018), the learner agent learns two basis policy functions $\{\mu_0, \mu_1\}$ from the two trajectory sets, where the policy μ_0 (or μ_1) approximates the expert policy $\pi_e^*(s_t, a_t, U = 0 \text{ or } 1)$. After clustering, the expert dataset τ_D can be written as $\{m_i, V_i\}$, where $m_i \in \{0, 1\}$ denotes the index of the basis policy function that generates the ith trajectory τ_i in τ_D. Using the data $\{m_i, V_i\}$, the learner agent can compute the emprical mean of the returns $\mathbb{E}\big[V|\mu_k\big]$ and the causal bounds on the true expected returns, as in (4). In addition, we can evaluate the true expected returns $\mathbb{E}\big[V|do(\mu_0 \text{ or } \mu_1)\big]$ of the expert policies $\pi_e^*(s_t, a_t, U = 0 \text{ or } 1)$ through simulations. These results are presented in Table 2. Observe that the basis policy function μ_1 achieves higher true expected accumulated rewards compared to the policy μ_0, i.e., $\mathbb{E}\big[V|do(\mu_1)\big] > \mathbb{E}\big[V|do(\mu_0)\big]$. Therefore, μ_1 is a better policy to select. However, if the basis policy function is selected based on its empirical expected returns $\mathbb{E}\big[V|\mu_k\big]$, the wrong policy

	$\mathbb{E}[V\|\mu_k]$	$\mathbb{E}[V\|do(\mu_k)]$	l_k	h_k
$do(\mu_0)$	0.479	-0.10	-0.89	0.407
$do(\mu_1)$	0.244	0.22	-2.69	0.407

Table 2: Empirical mean, true mean and causal bounds on the accumulated rewards.

μ_0 will be selected. Furthermore, the causal bounds (4) in Table 2 correctly characterize the range of the value of $\mathbb{E}\big[V|do(\mu_k)\big]$.

Having constructed the basis policy functions $\{\mu_k\}$, we then compared Algorithm 1 to a MAB method on the set of basis functions without causal bounds, and the direct imitation learning method in Torabi et al. (2018) that learns a single policy from the expert dataset τ_D and transfers this policy directly to the learner agent. The results are presented in Figure 2(b)subfigure. Observe that TL in the case of MAB without causal bounds (blue curve) or direct imitation learning (red curve) suffers large variance during online training. This is because both the policy π_{DI} obtained from τ_D using direct imitation learning and the basis policy function μ_0 selected by MAB are far from the optimal policy. Improving these suboptimal polices online generally results in higher variance during training compared to improving a good quality policy, e.g., the basis policy function μ_1. On the other hand, our proposed Algorithm 1 uses the causal bounds to reduce the variance of MAB exploration and finds the best basis policy function much faster than MAB without causal bounds. To elaborate, when the UCB on the expected return of the suboptimal basis policy function computed using (2) is over-optimistic, the causal bounds project this UCB to a tigher bound. This reduces the number of times the suboptimal basis policy function is explored. Consequently, Algorithm 1 (orange curve) can find the optimal policy within much fewer episodes compared to the other two methods and enjoys the lowest training variance, as shown in Figure 2(b)subfigure.

5. Conclusion

In this paper, we considered a transfer RL problem in continuous state and action spaces under unobserved contextual information. The goal was to use data generated by an expert agent that knows the context to learn an optimal policy for a learner agent that can not observe the same context, using only a few new data samples. To this date, such problems are typically solved using imitation learning that assumes that both the expert and learner agents have access to the same information. However, if the learner does not know the expert context, using expert data alone will result in a biased learner policy and will require many new data samples to improve. To address this challenge, in this paper, we proposed a MAB method that uses the expert data to train initial basis policies that serve as the arms in the MAB and to compute causal bounds on the accumulated rewards of these basis policies to reduce the MAB exploration variance and improve the learning rate. We provided numerical experiments on an autonomous driving example that showed that our proposed transfer RL method improved the learner's policy faster compared to imitation learning methods and enjoyed much lower variance during training.

Acknowledgments

This work is supported in part by AFOSR under award #FA9550-19-1-0169 and by NSF under award CNS-1932011.

References

André Barreto, Will Dabney, Rémi Munos, Jonathan J Hunt, Tom Schaul, Hado P van Hasselt, and David Silver. Successor features for transfer in reinforcement learning. In *Advances in neural information processing systems*, pages 4055–4065, 2017.

Pim de Haan, Dinesh Jayaraman, and Sergey Levine. Causal confusion in imitation learning. In *Advances in Neural Information Processing Systems*, pages 11698–11709, 2019.

Yan Duan, Marcin Andrychowicz, Bradly Stadie, Jonathan Ho, Jonas Schneider, Ilya Sutskever, Pieter Abbeel, and Wojciech Zaremba. One-shot imitation learning. In *Advances in Neural Information Processing Systems*, pages 1087–1098, 2017.

Jalal Etesami and Philipp Geiger. Causal transfer for imitation learning and decision making under sensor-shift. *arXiv preprint arXiv:2003.00806*, 2020.

Seyed Kamyar Seyed Ghasemipour, Richard Zemel, and Shixiang Gu. A divergence minimization perspective on imitation learning methods. In *Conference on Robot Learning*, pages 1259–1277. PMLR, 2020.

Assaf Hallak, Dotan Di Castro, and Shie Mannor. Contextual markov decision processes. *CoRR*, abs/1502.02259, 2015.

Karol Hausman, Yevgen Chebotar, Stefan Schaal, Gaurav Sukhatme, and Joseph J Lim. Multi-modal imitation learning from unstructured demonstrations using generative adversarial nets. In *Advances in Neural Information Processing Systems*, pages 1235–1245, 2017.

Jonathan Ho and Stefano Ermon. Generative adversarial imitation learning. In *Advances in neural information processing systems*, pages 4565–4573, 2016.

Arthur Juliani, Vincent-Pierre Berges, Esh Vckay, Yuan Gao, Hunter Henry, Marwan Mattar, and Danny Lange. Unity: A general platform for intelligent agents. *arXiv preprint arXiv:1809.02627*, 2018.

George Konidaris, Ilya Scheidwasser, and Andrew G Barto. Transfer in reinforcement learning via shared features. *The Journal of Machine Learning Research*, 13(1):1333–1371, 2012.

Alessandro Lazaric. Transfer in reinforcement learning: a framework and a survey. In *Reinforcement Learning*, pages 143–173. Springer, 2012.

Alessandro Lazaric, Marcello Restelli, and Andrea Bonarini. Transfer of samples in batch reinforcement learning. In *Proceedings of the 25th international conference on Machine learning*, pages 544–551, 2008.

Lisa Lee, Benjamin Eysenbach, Emilio Parisotto, Eric Xing, Sergey Levine, and Ruslan Salakhutdinov. Efficient exploration via state marginal matching. *arXiv preprint arXiv:1906.05274*, 2019.

Yunzhu Li, Jiaming Song, and Stefano Ermon. Infogail: Interpretable imitation learning from visual demonstrations. In *Advances in Neural Information Processing Systems*, pages 3812–3822, 2017.

Aniruddh Raghu, Matthieu Komorowski, Leo Anthony Celi, Peter Szolovits, and Marzyeh Ghassemi. Continuous state-space models for optimal sepsis treatment-a deep reinforcement learning approach. *arXiv preprint arXiv:1705.08422*, 2017.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Seyed Kamyar Seyed Ghasemipour, Shixiang (Shane) Gu, and Richard Zemel. Smile: Scalable meta inverse reinforcement learning through context-conditional policies. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/2b8f621e9244cea5007bac8f5d50e476-Paper.pdf.

Matthew E Taylor and Peter Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10(7), 2009.

Matthew E Taylor, Peter Stone, and Yaxin Liu. Transfer learning via inter-task mappings for temporal difference learning. *Journal of Machine Learning Research*, 8(Sep):2125–2167, 2007.

Andrea Tirinzoni, Mattia Salvini, and Marcello Restelli. Transfer of samples in policy search via multiple importance sampling. In *International Conference on Machine Learning*, pages 6264–6274, 2019.

Faraz Torabi, Garrett Warnell, and Peter Stone. Behavioral cloning from observation. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, pages 4950–4957, 2018.

Lantao Yu, Tianhe Yu, Chelsea Finn, and Stefano Ermon. Meta-inverse reinforcement learning with probabilistic context variables. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/30de24287a6d8f07b37c716ad51623a7-Paper.pdf.

Junzhe Zhang and Elias Bareinboim. Transfer learning in multi-armed bandits: A causal approach. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 1340–1346, 2017.

Junzhe Zhang and Elias Bareinboim. Bounding causal effects on continuous outcomes. 2020.

Junzhe Zhang, Daniel Kumor, and Elias Bareinboim. Causal imitation learning with unobserved confounders. *Advances in Neural Information Processing Systems*, 33, 2020.

Yan Zhang and Michael M Zavlanos. Transfer reinforcement learning under unobserved contextual information. In *2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPS)*, pages 75–86. IEEE, 2020.

Yijie Zhou, Yan Zhang, Xusheng Luo, and Michael M Zavlanos. Human-in-the-loop robot planning with non-contextual bandit feedback. *arXiv preprint arXiv:2011.01793*, 2020.

Yuke Zhu, Roozbeh Mottaghi, Eric Kolve, Joseph J Lim, Abhinav Gupta, Li Fei-Fei, and Ali Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3357–3364. IEEE, 2017.