

# Model-Based Transfer Learning for Contextual Reinforcement Learning

**Jung-Hoon Cho**  
MIT  
[jhooncho@mit.edu](mailto:jhooncho@mit.edu)

**Vindula Jayawardana**  
MIT  
[vindula@mit.edu](mailto:vindula@mit.edu)

**Sirui Li**  
MIT  
[siruili@mit.edu](mailto:siruili@mit.edu)

**Cathy Wu**  
MIT  
[cathywu@mit.edu](mailto:cathywu@mit.edu)

## Abstract

Deep reinforcement learning (RL) is a powerful approach to complex decision making. However, one issue that limits its practical application is its brittleness, sometimes failing to train in the presence of small changes in the environment. Motivated by the success of zero-shot transfer—where pre-trained models perform well on related tasks—we consider the problem of selecting a good set of training tasks to maximize generalization performance across a range of tasks. Given the high cost of training, it is critical to select training tasks strategically, but not well understood how to do so. We hence introduce Model-Based Transfer Learning (MBTL), which layers on top of existing RL methods to effectively solve contextual RL problems. MBTL models the generalization performance in two parts: 1) the performance set point, modeled using Gaussian processes, and 2) performance loss (generalization gap), modeled as a linear function of contextual similarity. MBTL combines these two pieces of information within a Bayesian optimization (BO) framework to strategically select training tasks. We show theoretically that the method exhibits sublinear regret in the number of training tasks and discuss conditions to further tighten regret bounds. We experimentally validate our methods using urban traffic and standard continuous control benchmarks. The experimental results suggest that MBTL can achieve up to 50x improved sample efficiency compared with canonical independent training and multi-task training. Further experiments demonstrate the efficacy of BO and the insensitivity to the underlying RL algorithm and hyperparameters. This work lays the foundations for investigating explicit modeling of generalization, thereby enabling principled yet effective methods for contextual RL. Code is available at <https://github.com/jhoon-cho/MBTL/>.

**Brittleness in the context of machine learning**, especially in deep reinforcement learning (RL) and other AI systems, refers to the tendency of a model or algorithm to perform poorly when faced with small changes, perturbations, or variations in the environment or input data. This lack of robustness can make AI systems unreliable in real-world scenarios where conditions are often unpredictable and dynamic.

## 1 Introduction

Deep reinforcement learning (DRL) has made remarkable strides in addressing complex problems across various domains [29, 36, 1, 4, 11, 13, 27]. Despite these successes, DRL algorithms often exhibit brittleness when exposed to small variations like different number of lanes, weather conditions, or flow density in traffic benchmarks [20], significantly limiting their scalability and generalizability [19]. Such variations can be modeled using the framework of contextual Markov Decision Processes (CMDP), where task variations can be parameterized within a context space [15, 32, 6].

There are two predominant solution modalities for CMDPs [22]: independent training and multi-task training. Independent training constructs a separate model for each task variant (say,  $N \gg 1$ ), which

is compute-intensive. At the other extreme, multi-task training constructs a single “universal” policy, and thus can be more compute-efficient, but suffers from model capacity and negative transfer issues [21, 44, 2, 41]. There is thus a need for more reliable training methodologies for generalization across tasks variants. In this work, we consider training an intermediate set of  $K$  models, where  $N > K > 1$ , in an effort to balance performance and efficiency; we refer to this strategy as *multi-policy training*.

**A note on terminology.** For brevity, we refer to *task variants* as *tasks* in the remainder of this article. We also use the language of *task* and *context* interchangeably. We emphasize that this work focuses on within-domain generalization (e.g., traffic signal control for intersection scenario variants) rather than across-domain generalization (e.g., distinct traffic control tasks). Additionally, it is crucial to differentiate between *training* reliability, which concerns the ability to reliably train models across tasks, and *model* reliability (or robustness), which concerns the resistance of a trained model to differences in tasks. This article is concerned with training reliability.

We build upon zero-shot transfer, a widely-used practical technique which directly applies a policy trained in one context (source task) to another (target task) without adaptation. This experiences performance degradation, also called *generalization gap* [17, 22]. For instance, Figure 1 depicts that the performance degrades as the target task diverges from the source task, corresponding to an increasing generalization gap. Nonetheless, leveraging the notion that training is expensive but zero-shot transfer is cheap, we are interested in optimally selecting a set of source (training) tasks, such that the generalization performance on the target range of tasks is maximized. In this article, we strategically select source tasks by explicitly modeling the generalization performance to estimate the value of training a new source task.

The main contributions of this work are:

- We introduce *Model-Based Transfer Learning (MBTL)*, a novel framework for studying generalization in reinforcement learning (Figure 2). To the best of our knowledge, this is the first work to explicitly model generalization performance for contextual RL. As such, our work opens the door for further investigation into reliable model-based methodologies for contextual RL.
- We provide theoretical analysis for the sublinear regret of MBTL and give conditions for achieving tighter regret bounds.
- We empirically validate our methods in urban traffic and standard continuous control benchmarks for contextual RL, observing up to 50x improvements in sample efficiency. We further include ablations on the components of the algorithm.

The remainder of the paper is organized as follows. After introducing notation in Section 2, we formally define the problem in Section 3. A key contribution of our work is the introduction of a Gaussian process model acquisition function specifically tailored to the source task selection problem, which is detailed in Section 4. In Section 4.3, we provide a theoretical analysis of the regret bounds of our method, followed by an empirical evaluation across diverse applications in Section 5.

## 2 Preliminaries and notation

**Contextual MDP.** A standard MDP is defined by the tuple  $M = (S, A, P, R, \rho)$  where  $S$  represents the state space,  $A$  is the action space,  $P$  denotes the transition dynamics,  $R$  is the reward function, and  $\rho$  is the distribution over initial states [43]. A contextual MDP (CMDP), denoted by  $\mathcal{M} = (S, A, P_x, R_x, \rho_x)_{x \in X}$ , is a collection of context-MDPs  $\mathcal{M}_x$  parameterized by a context variable  $x$  within a context space  $X$  (assumed bounded). The context variable  $x$  can influence dynamics, rewards, and initial state distributions [15, 32, 6]. We define source task performance  $J(\pi_x, x; \text{Alg})$  as follows: we train a policy  $\pi_x$  on a task with the context  $x \in X$  using RL algorithm Alg (e.g., PPO, SAC) and evaluate the policy by the expected return in the same MDP  $\mathcal{M}_x$  with

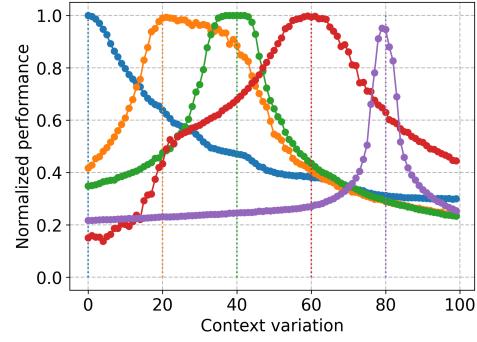


Figure 1: Example generalization gap depicted for Cartpole CMDP. The solid lines show the true zero-shot transfer generalization performance across contexts. Source tasks are indicated by dotted lines.

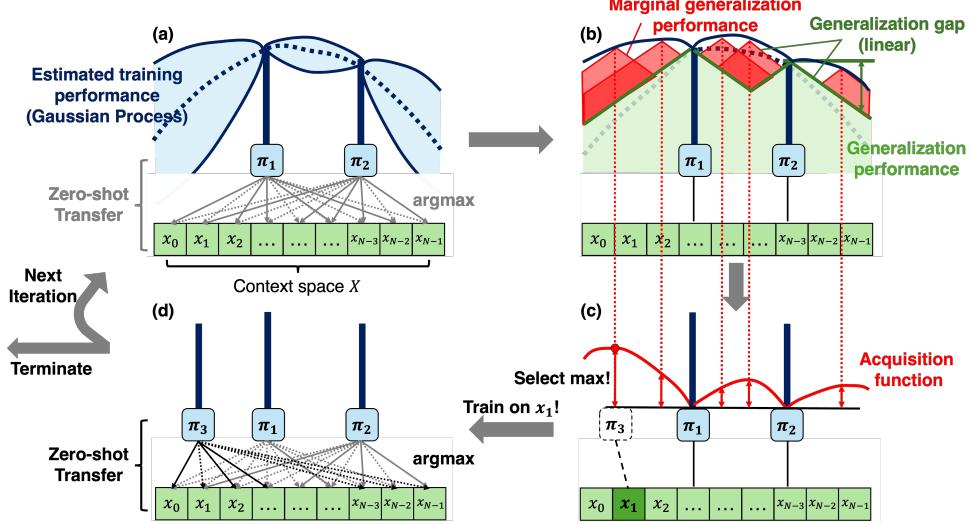


Figure 2: **Overview illustration for Model-based Transfer Learning.** (a) Gaussian process regression is used to estimate the training performance across tasks using existing policies; (b) marginal generalization performance (red area) is calculated using upper confidence bound of estimated training performance, generalization gap, and generalization performance; (c) selects the next training task that maximizes the acquisition function (marginal generalization performance); (d) once the selected task is trained, calculate generalization performance using zero-shot transfer.

context  $x$ . For brevity, we will write it as  $J(\pi_x, x)$ . We distinguish between estimated values  $\hat{J}(x)$  and observed outcomes  $J(\pi_x, x)$ , with the latter measured after training and evaluation.

**Generalization gap via zero-shot transfer.** Consider zero-shot transfer from the trained policy  $\pi_x$  from a source task (context-MDP) to solve another target task (context-MDP) with the context  $x' \in X$  in the CMDP. Zero-shot transfer involves applying a policy trained on a source task  $\mathcal{M}_x$  to a different target task  $\mathcal{M}_{x'}$ , with  $x, x' \in X$ . We observe the *generalization performance*, denoted by  $J(\pi_x, x')$ , by evaluating the target task  $x'$  based on the policy trained using source task  $x$  via zero-shot generalization. We define the generalization gap as the absolute performance difference in average reward when transferring from source task  $x$  to target task  $x'$ :

$$\underbrace{\Delta J(\pi_x, x')}_{\text{Generalization gap}} = \underbrace{J(\pi_x, x)}_{\text{Source task performance}} - \underbrace{J(\pi_x, x')}_{\text{Generalization performance}}. \quad (1)$$

### 3 Problem formulation

**Sequential source task selection problem.** The selection of source MDPs from the CMDPs is key to solving the overall CMDP [3]. We therefore introduce the *sequential source task selection (SSTS) problem*, which seeks to maximize the expected performance across a dynamically selected set of tasks. This problem is cast as a sequential decision problem, in which the selection of tasks is informed through feedback from the observed task performance of the tasks selected and trained thus far. The notation  $x_k$  indicates the selected source task at the  $k$ -th transfer step, where  $k$  ranges from 1 to  $K$ . For brevity, we will denote  $\pi_{x_k}$  as  $\pi_k$ . We denote the sequence  $x_1, x_2, \dots, x_k$  by  $x_{1:k}$  and  $\pi_1, \pi_2, \dots, \pi_k$  by  $\pi_{1:k}$ .

**Definition 1** (Sequential Source Task Selection Problem). *This problem seeks to optimize the expected generalization performance across a CMDP  $\mathcal{M}_{x' \in X}$  by selecting a task  $x \in X$  at each training stage. Specifically, at each selection step  $k$ , we wish to choose a distinct task  $x_k$  such that the expected cumulative generalization performance is maximized. This can be expressed by keeping track at each step, which policy to use for which task, and the corresponding generalization performance. Upon training the policy  $\pi_{x_k}$  for source task  $x_k$ , the cumulative generalization performance, which we abuse notation to denote as  $J(\pi_{1:k}, \cdot) = J(\pi_{x_k}, \cdot; \pi_{1:k-1})$ . Formally, this can be recursively defined based on previous observations  $\{J(\pi_1, x), \dots, J(\pi_{k-1}, x)\}$  for all  $x \in X$  as follows:*

$$J(\pi_{x_k}, x'; \pi_{1:k-1}) = \max(J(\pi_k, x'), J(\pi_{1:k-1}, x')) \quad \forall x' \in X \quad \text{if } k > 1. \quad (2)$$

And  $J(\pi_{1:1}, x) \equiv J(\pi_1, x)$ . Then, the overall sequential decision problem can be written as:

$$\max_{x_k} V(x_k; \pi_{1:k-1}) = \max_{x_k} \mathbb{E}_{x' \sim \mathcal{U}(X)} [J(\pi_{x_k}, x'; \pi_{1:k-1})] \quad s.t. \quad x_k \in X \setminus x_{1:k-1}. \quad (3)$$

The state at each step  $k$  is defined by the best generalization performance for each task, achieved by policies trained in earlier stages, represented as  $J(\pi_{1:k-1}, x')$  for each target task  $x'$ . The action at each step is choosing a new task  $x_k$ . In general, SSTS exhibits stochastic transitions, for example due to randomness in RL training. For simplicity, in this work, we assume deterministic transitions; that is, training context-MDP  $x$  will always yield the same performance  $J(\pi_x, x)$  and generalization gap  $\Delta J(\pi_x, x')$ ,  $\forall x' \in X$ . The problem's maximum horizon is defined by  $|X|$ , but can be terminated early if conditions are met (e.g., performance level, training budget).

## 4 Model-Based Transfer Learning (MBTL)

In this section, we introduce an algorithm called Model-based Transfer Learning to solve the SSTS problem. MBTL models the generalization performance in two parts: 1) the performance set point, modeled using Gaussian processes, and 2) performance loss (generalization gap), modeled as a linear function of contextual similarity. MBTL combines these two pieces of information within a Bayesian optimization (BO) framework to sequentially select training tasks that maximize generalization performance. The high-level pseudo-code of the MBTL algorithm is given below.

### MBTL Pseudo-code

- ```

1: Input: Task set, Gaussian Process (GP) model, and training budget
2: while training budget remains do
3:   Select next source task  $\leftarrow$  Task with the highest acquisition value based on estimated training
      performance, generalization gap, and generalization performance, determined by the current GP
      model and tailored acquisition function
4:   Train a policy on the selected source task using reinforcement learning, then evaluate its training
      performance
5:   Update the GP model with the selected source task training performance to estimate the training
      performance of tasks in the task set
6: end while
7: Output: Set of policies that maximizes generalized performance across task set

```

### 4.1 Modeling assumptions

We consider a task set  $X$  that is continuous and the performance function  $J(\pi, x), V(x)$  for a policy  $\pi$  to be smooth over the task space  $X$ . In practice, such as control systems, tasks often vary continuously and smoothly rather than abruptly. For example, adjusting the angle of a robotic arm by a small amount typically results in a small change in the system and optimal action.

Inspired by the empirical generalization gap performance as observed in Figure 1, we estimate the generalization gap  $\Delta\hat{J}(\pi_x, x')$  with a linear function.

**Assumption 1** (Linear generalization gap). *A linear function is used to model the generalization gap function, formally  $\Delta\hat{J}(\pi_x, x') \simeq \theta|x - x'|$ , where  $\theta$  is the slope of the linear function and  $x$  and  $x'$  are the context of the source task and target task, respectively.*

The relaxation of this assumption can yield additional efficiency benefits but also adds modeling complexity, and thus is an interesting direction for future work.

### 4.2 Bayesian optimization

Bayesian optimization (BO) is a powerful strategy for finding the global optimum of an objective function when obtaining the function is costly, or the function itself lacks a simple analytical form [31, 7]. BO integrates prior knowledge with observations to efficiently decide the next task to train by using the acquisition function. MBTL is a BO method which optimizes for promising source tasks by leveraging Assumption 1 in its acquisition function. The role of BO is to approximate  $V(x_k; \pi_{1:k-1})$  (see Equation 3) using the data acquired thus far. The next source task  $x_k$  is then selected using this estimate.

**Gaussian process (GP) regression.** Within the framework of BO, we model the source training performance  $\hat{J}(\pi_x, x) \forall x \in X \setminus x_{1:k}$  using Gaussian process (GP) regression. Specifically, the function  $\hat{J}(\pi_x, x)$  is assumed to follow a GP ( $\hat{J}(\pi_x, x) \sim \mathcal{GP}(\mathbb{E}[\hat{J}(\pi_x, x)], k(x, \tilde{x}))$ ), where  $k(x, \tilde{x})$  is the covariance function, representing the expected product of deviations of  $\hat{J}(\pi_x, x)$  and  $\hat{J}(\pi_{\tilde{x}}, \tilde{x})$  from their respective means. Let  $D_{k-1}$  denote the data observed up to iteration  $k-1$ , consisting of the pairs  $\{(x_i, J(\pi_i, x_i))\}_{i=1,\dots,k-1}$ . The estimated performance  $\hat{J}_k$  after querying  $k-1$  samples is updated as more samples are obtained. The posterior prediction of  $\hat{J}_k$  at a new point  $x$ , given the data  $D_{k-1}$  and previous inputs  $x_{1:k-1}$ , is normally distributed as  $P(\hat{J}_k | D_{k-1}) = \mathcal{N}(\mu_k(x), \sigma_k^2(x))$ .  $\mu_k(x)$  and  $\sigma_k^2(x)$  are defined as  $\mu_k(x) = \mathbb{E}[\hat{J}(\pi_x, x)] + \mathbf{k}^\top (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}$  and  $\sigma_k^2(x) = k(x, x) - \mathbf{k}^\top (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{k}$ , with  $\mathbf{k}$  being the vector of covariances between  $x$  and each  $x_i$  in the observed data, i.e.,  $\mathbf{k} = [k(x, x_1), \dots, k(x, x_{k-1})]$ , and  $\mathbf{K}$  is the covariance matrix for the observed inputs, defined as  $\mathbf{K} = [k(x_i, x_j)]_{1 \leq i, j \leq k-1}$ . This enables the GP to update its beliefs about the posterior prediction with every new observation, progressively improving the estimation.

**Acquisition function.** The acquisition function plays a critical role in BO by guiding the selection of the next source training task. At each decision step  $k$ , the task  $x_k$  is chosen by maximizing the acquisition function, as denoted by  $x_k = \arg \max_x a(x; x_{1:k-1})$ . One effective strategy employed in the acquisition function is the upper confidence bound (UCB) acquisition function, which considers the trade-off between the expected performance of a task based on the current task (exploitation) and the measure of uncertainty associated with the task's outcome (exploration) [40]. Especially in our case, the acquisition function can be designed as UCB function subtracted by generalization gap and so-far best performance. It is defined as follows:

$$a(x; x_{1:k-1}) = \mathbb{E}_{x' \in X} [[\mu_{k-1}(x) + \beta_k^{1/2} \sigma_{k-1}(x) - \Delta \hat{J}(\pi_x, x') - J(\pi_{1:k-1}, x')]_+] \quad (4)$$

where  $[\cdot]_+$  represents  $\max(\cdot, 0)$  and we can use various forms of  $\beta_k$ , which is the trade-off parameter between exploitation and exploration.

### 4.3 Regret analysis

We use regret to quantify the effectiveness of our source task selection based on BO. Specifically, we define regret at iteration  $k$  as  $r_k = \hat{V}(x_k^*; \pi_{1:k-1}) - \hat{V}(x_k; \pi_{1:k-1})$ , where  $\hat{V}(x_k^*; \pi_{1:k-1})$  represents the maximum generalization performance achievable across all tasks, and  $\hat{V}(x_k; \pi_{1:k-1})$  is the performance at the current task selection  $x_k$ . Consequently, the cumulative regret after  $K$  iterations is given by  $R_K = \sum_{k=1}^K r_k$ , summing the individual regrets over all iterations. Following the framework presented by Srinivas et al. [40], our goal is to establish that this cumulative regret grows sublinearly with respect to the number of iterations. Mathematically, we aim to prove that  $\lim_{K \rightarrow \infty} \frac{R_K}{K} = 0$ , indicating that, on average, the performance of our strategy approaches the optimal performance as the number of iterations increases.

**Regret of MBTL.** Having established the general framework for regret analysis, we now turn our attention to the specific regret properties of our MBTL algorithm. To analyze the regret of MBTL, consider the scaling factor for the UCB acquisition function given by  $\beta_k = 2 \log(|X| \pi^2 k^2 / 6\delta)$  in Equation (4). It is designed to achieve sublinear regret with high probability, aligning with the theoretical guarantees outlined in Theorem 1 and 5 from [40].

**Theorem 1** (Sublinear Regret). *Given  $\delta \in (0, 1)$ , and with the scaling factor  $\beta_k$  as defined, the cumulative regret  $R_K$  is bounded by  $\sqrt{KC_1\beta_K\gamma_K}$  with a probability of at least  $1 - \delta$ . The formal expression of this probability is  $\Pr[R_K \leq \sqrt{KC_1\beta_K\gamma_K}] \geq 1 - \delta$ , where  $C_1 := \frac{8}{\log(1+\sigma^{-2})} \geq 8\sigma^2$  and  $\gamma_K = \mathcal{O}(\log K)$  for the squared exponential kernel.*

**Impact of search space elimination.** In this section, we demonstrate that strategic reduction of the possible sets, guided by insights from previous task selections or source task training performance, leads to significantly tighter regret bounds than Theorem 1. By focusing on the most promising regions of the task space, our approach enhances learning efficiency and maximizes the policy's performance and applicability. Given the generalization gap observed in Figure 1, we observe that performance loss decreases as the context similarity increases. We model the degradation from the source task using a linear function in Assumption 1. Training on the source task can solve a significant portion of the remaining tasks. Our method progressively eliminates partitions of the task

space at a certain rate with each iteration. If the source task selected in the previous steps could solve the remaining target task sufficiently, we can eliminate the search space at a desirable rate. Formally, we can define the search space at  $k$ -th iteration as follows:

**Definition 2** (Search space). *We define the search space  $X_k$  at iteration  $k$  as a subset of  $X$ , with each element  $x' \in X_k$ , such that  $J(\pi_{1:k-1}, x') \leq \hat{J}(\pi_{x_k}, x') - \Delta J(\pi_{x_k}, x')$ .*

Given the generalization gap observed in Figure 1, we model the degradation from the source task using a linear function in Assumption 1. While the figure might not strictly appear linear, the linear approximation simplifies analysis and is supported by empirical observations. Training on the source task can solve a significant portion of the remaining tasks. Our method progressively eliminates partitions of the task space at a certain rate with each iteration. If the source task selection in the previous step sufficiently addresses the remaining target tasks, we can reduce the search space at a desirable rate. Consequently, at each step, we effectively focus on a reduced search space.

We leverage the reduced uncertainty in well-sampled regions to tighten the regret bound while slightly lowering the probability  $\delta$  in Theorem 1. For the regret analysis, we propose the following theorem based on the generalization of Lemma 5.2 and 5.4 in [40] to the eliminated search space.

**Theorem 2.** *For a given  $\delta' \in (0, 1)$  and scaling factor  $\beta_k = 2 \log(|X| \pi^2 k^2 / 6\delta)$ , the cumulative regret  $R_K$  is bounded by  $\sqrt{C_1 \beta_K \gamma_K \sum_{k=1}^K \left( \frac{|X_k|}{|X|} \right)^2}$  with probability at least  $1 - \delta'$ .*

Here,  $|X|$  denotes the cardinality of the set  $X$ , the number of elements in  $X$ . Theorem 2 matches the Theorem 1 when  $X_k = X$  for all  $k$ . This theorem implies that regret has a tighter or equivalent bound if we can design the smaller search space instead of searching the whole space. The comprehensive proof is provided in Appendix A.3.1.

Here are some examples of restricted search space: If we consider an example where  $|X_k| = \frac{1}{\sqrt{k}} |X|$ , the regret can be bounded tighter than that of Theorem 1.

**Corollary 2.1.** *Consider  $|X_k| = \frac{1}{\sqrt{k}} |X|$ . The regret bound would be  $R_K \leq \sqrt{C_1 \beta_K \gamma_K \log K}$  with a probability of at least  $1 - \delta'$ .*

In cases where the search space is defined using MBTL-GS, the largest segment's length would reduce geometrically, described by  $|X_k| \leq 2^{-\lfloor \log_2 k \rfloor} |X|$ .

**Corollary 2.2.** *The regret bound for the  $|X_k| \leq 2^{-\lfloor \log_2 k \rfloor} |X|$  would be  $R_K \leq \sqrt{C_1 \beta_K \gamma_K \pi^2 / 6}$  with a probability of at least  $1 - \delta'$ .*

Proofs for Corollaries 2.1 and 2.2 are provided in Appendix A.3.2 and A.3.3, respectively. Based on our experiments presented in Section 5, the rate of elimination of the largest segment for MBTL is shown in Figure 3.

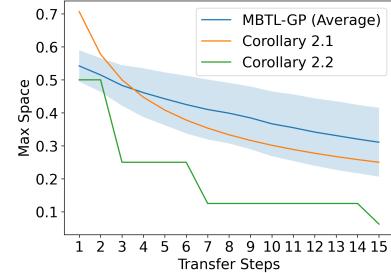


Figure 3: Empirical results of the restriction of search space by MBTL compared to two examples from Corollaries 2.1 and 2.2.

## 5 Experiments and analysis

### 5.1 Setup

Our experiments consider CMDPs that span standard and real-world benchmarks. In particular, we consider standard continuous control benchmarks from the CARL library [6]. In addition, we study problems from RL for intelligent transportation systems, using [47] to model the CMDPs. Surprisingly, despite the relatively low complexity of the CMDPs considered, standard deep RL algorithms appear to struggle to solve the tasks.

**Baselines.** We consider two types of baselines when evaluating our proposed algorithm: canonical and multi-policy. The canonical baselines are selected to validate multi-policy training; the multi-policy training baselines are heuristic methods designed to validate the Bayesian optimization approach. The canonical baselines include: (1) **Independent training**, which involves independently training separate models on each task; and (2) **Multi-task RL**, where a single “universal” context-conditioned

Table 1: Comparative performance of different methods on traffic CMDPs

| Benchmark (CMDP)                |                   | Baselines   |               | Sequential Baselines |               |               | MBTL          | Oracle     |
|---------------------------------|-------------------|-------------|---------------|----------------------|---------------|---------------|---------------|------------|
| Domain                          | Context Variation | Independent | Multi-task    | Random               | Equidistant   | Greedy        | Ours          | Sequential |
| <b>Number of Trained Models</b> |                   | <i>N</i>    | 1             | <i>k</i>             | <i>K</i>      | <i>k</i>      | <i>k</i>      | <i>N</i>   |
| Traffic Signal                  | Inflow            | 0.8646      | 0.8319        | 0.8457               | <b>0.8700</b> | 0.8496        | <b>0.8673</b> | 0.8768     |
| Traffic Signal                  | Speed Limit       | 0.8857      | 0.6083        | <b>0.8821</b>        | <b>0.8858</b> | <b>0.8862</b> | <b>0.8854</b> | 0.8876     |
| Eco-driving                     | Penetration Rate  | 0.5260      | 0.1945        | 0.5959               | 0.5934        | 0.5827        | <b>0.6323</b> | 0.6660     |
| Eco-driving                     | Inflow            | 0.4061      | 0.2229        | 0.4774               | 0.4705        | 0.4673        | <b>0.5108</b> | 0.5528     |
| Eco-driving                     | Green Phase       | 0.3850      | 0.4228        | 0.4406               | <b>0.4557</b> | 0.4431        | <b>0.4700</b> | 0.5027     |
| AA-Ring-Acc                     | Hold Duration     | 0.8362      | <b>0.9209</b> | 0.8924               | <b>0.9057</b> | 0.8776        | <b>0.9242</b> | 0.9552     |
| AA-Ring-Vel                     | Hold Duration     | 0.9589      | 0.972         | 0.9785               | 0.9772        | 0.9807        | <b>0.9816</b> | 0.9822     |
| AA-Ramp-Acc                     | Hold Duration     | 0.4276      | 0.5158        | <b>0.6050</b>        | 0.5956        | <b>0.6143</b> | <b>0.6318</b> | 0.7111     |
| AA-Ramp-Vel                     | Hold Duration     | 0.5473      | 0.5034        | 0.669                | <b>0.6787</b> | 0.5907        | <b>0.7182</b> | 0.7686     |
| <b>Average</b>                  |                   | 0.6778      | 0.6017        | 0.7312               | 0.7354        | 0.722         | 0.7559        | 0.7844     |

†Higher the better. Bold values represent the highest value(s) within the statistically significant range for each CMDP, excluding the oracle. Detailed results with variance for each method are provided in Appendix A.4.2.

‡AA: Advisory autonomy benchmark, Ring: Single lane ring, Ramp: Highway ramp, Acc: Acceleration guidance, Vel: Speed guidance.

policy is trained for all tasks. The multi-policy baselines include: (3) **Random selection**, where each training task is chosen uniformly at randomly; (4) **Equidistant strategy**, which selects training tasks by equally subdividing the context space based on a given training budget, and then trains them in lexicographical order; (5) **Greedy strategy**, which greedily selects the next source task based on Assumption 1 and fixed training performance; and (6) **Sequential Oracle transfer**, which has access to generalized performance corresponding to policies for all tasks (including those not yet selected) and uses that information to greedily select the best source task at each step.

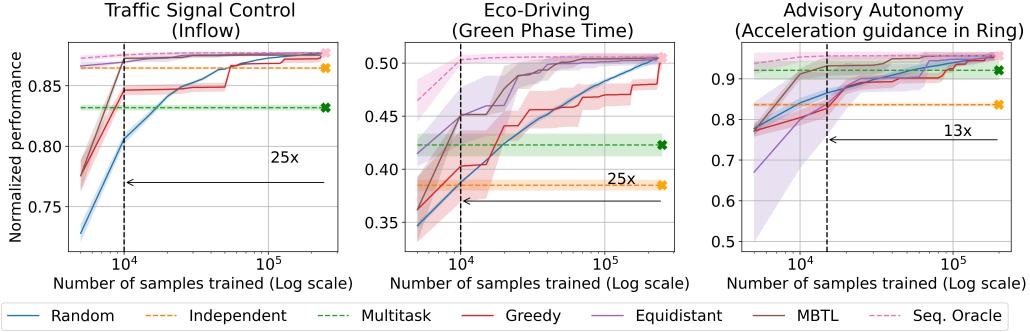
**Proposed method.** We evaluate MBTL with the scaling factor  $\beta_k = 2 \log(|X|\pi^2 k^2/6\delta)$ .

**DRL algorithms and performance measure.** We utilize Deep Q-Networks (DQN) for discrete action spaces [29] and Proximal Policy Optimization (PPO) for continuous action spaces [35]. We evaluate our method by the average performance across all  $N$  target tasks after training up to  $K=15$  source tasks or the number of source tasks needed to achieve a certain level of performance. We employ min-max normalization of the rewards for each task, and we provide comprehensive details about our model in Appendix A.4.1.

## 5.2 Traffic benchmark experiments

We consider three traffic control domains. First, while most traffic lights still operate on fixed schedules, RL can be used to design adaptive (1) **Traffic signal control** to optimize traffic [9, 23]. However, considering that every intersection is different, challenges persist in generalizing across intersection configurations [19]. Given the significant portion of greenhouse gas emissions in the United States due to transportation [12], the second traffic domain is (2) **Dynamic eco-driving at signalized intersections**, which concerns learning energy-efficient driving strategies in urban settings. DRL-based eco-driving strategies have been developed [14, 45, 18] but still experience difficulties in generalization. Our final traffic domain is (3) **Advisory autonomy**, in which real-time speed or acceleration advisories guide human drivers to act as vehicle-based traffic controllers in mixed traffic environments [39, 8, 16]. The context space  $X$  is discretized into  $N = \{50, 50, 40\}$  contexts for the three domains, respectively. In Appendix A.4, we provide details about our experiments.

**Results.** Table 1 and Figure 4 summarize the results. Notably, the Oracle far outperforms the standard baselines (Independent and Multi-task), indicating the potential for transfer learning and intelligent training of multiple models, respectively. MBTL rapidly approaches the Oracle, within  $\approx 10$  transfer steps, indicating that the Gaussian process effectively models the training performance and linear generalization gap models generalization performance. It is important to note that multi-task RL studies commonly consider Independent training as a strong baseline due to its avoidance of negative transfer and other training instability issues. Indeed, Independent often (but not always) outperforms Multi-task in our experiments. Yet, our experiments show that MBTL consistently outperforms both Independent and Multi-task and matches their performance with **5-25x better sample efficiency**. Among the sequential baselines, MBTL consistently outperforms the heuristic baselines, indicating the value of adaptively selecting source tasks based on feedback. The sequential baselines, such as random, equidistant, and greedy strategy, also generally outperform Independent and Multi-task,

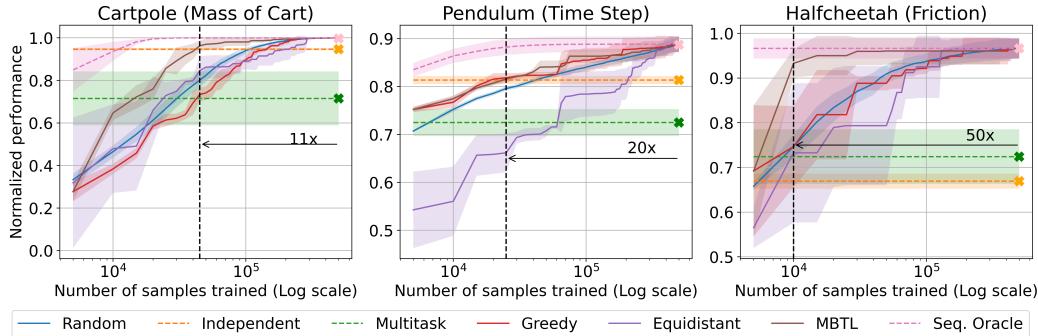


**Figure 4: Traffic CMDP results.** Method comparison of normalized performance over  $N$  tasks. MBTL efficiently selects source training tasks. The black dotted line indicates the first training step within MBTL that exceeds both independent and multi-task baselines, with up to 25x fewer samples needed.

Table 2: Comparative performance of different methods on standard control CMDPs

| Benchmark (CMDP)         |                   | Baselines     |            |        | Sequential Baselines |               |               | MBTL       | Oracle |
|--------------------------|-------------------|---------------|------------|--------|----------------------|---------------|---------------|------------|--------|
| Domain                   | Context Variation | Independent   | Multi-task | Random | Equidistant          | Greedy        | Ours          | Sequential |        |
| Number of Trained Models |                   | $N$           | 1          | $k$    | $K$                  | $k$           | $k$           | $N$        |        |
| Pendulum                 | Length            | 0.7383        | 0.6830     | 0.7270 | 0.7092               | 0.7327        | <b>0.7697</b> | 0.7969     |        |
| Pendulum                 | Mass              | 0.6237        | 0.5793     | 0.6329 | 0.6092               | 0.6408        | <b>0.6827</b> | 0.7132     |        |
| Pendulum                 | Timestep          | 0.8135        | 0.7247     | 0.7989 | 0.7541               | <b>0.8177</b> | 0.8141        | 0.8801     |        |
| Cartpole                 | Mass of Cart      | <b>0.9466</b> | 0.7153     | 0.7221 | 0.7516               | 0.6501        | 0.8212        | 0.9838     |        |
| Cartpole                 | Length of Pole    | <b>0.9110</b> | 0.5441     | 0.8121 | 0.8428               | 0.8217        | <b>0.9124</b> | 0.9875     |        |
| Cartpole                 | Mass of Pole      | <b>0.9560</b> | 0.6073     | 0.8858 | 0.7909               | 0.8744        | 0.9351        | 1.0000     |        |
| BipedalWalker            | Gravity           | 0.9281        | 0.7898     | 0.9330 | <b>0.9494</b>        | 0.9359        | <b>0.9393</b> | 0.9674     |        |
| BipedalWalker            | Friction          | 0.9317        | 0.9051     | 0.965  | 0.9664               | 0.9645        | <b>0.9713</b> | 0.9778     |        |
| BipedalWalker            | Scale             | 0.8694        | 0.7452     | 0.8605 | 0.8496               | <b>0.8792</b> | <b>0.8886</b> | 0.9107     |        |
| HalfCheetah              | Gravity           | 0.6679        | 0.6292     | 0.8542 | 0.8634               | 0.8663        | <b>0.9073</b> | 0.9544     |        |
| HalfCheetah              | Friction          | 0.6693        | 0.7242     | 0.8567 | 0.8703               | 0.8591        | <b>0.9274</b> | 0.9663     |        |
| HalfCheetah              | Stiffness         | 0.6561        | 0.7007     | 0.8533 | 0.7817               | 0.8785        | <b>0.9146</b> | 0.9674     |        |
| <b>Average</b>           |                   | 0.8093        | 0.6957     | 0.8251 | 0.8116               | 0.8267        | 0.8736        | 0.9255     |        |

<sup>†</sup>Higher the better. Bold values represent the highest value(s) within the statistically significant range for each CMDP, excluding the oracle. Detailed results with variance for each method are provided in Appendix A.4.2.



**Figure 5: Continuous control CMDP results.** Method comparison of normalized performance over  $N$  tasks. The black dotted line indicates the first training step within MBTL that exceeds both independent and multi-task baselines, with up to 50x fewer samples needed.

indicating the general value of multi-policy training for solving CMDPs. More results are provided in Appendix A.4.

### 5.3 Continuous control benchmark experiments

To probe the generality of MBTL, we utilize context-extended versions of standard RL environments from CARL benchmark library [6] to evaluate our methods under varied contexts. For the Cartpole

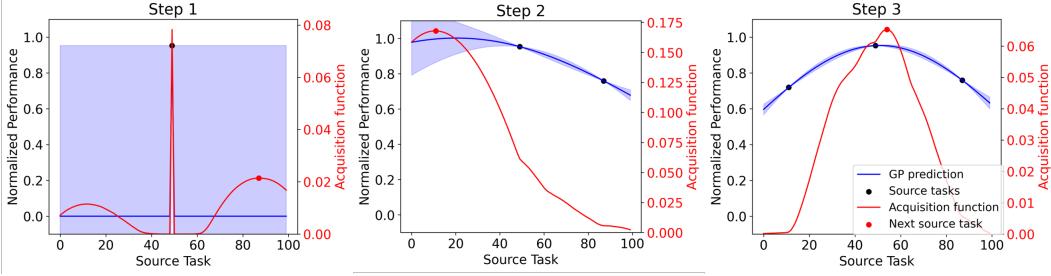


Figure 6: The GP sequentially updates estimates of the performance function (blue) based on previously trained models. Then, MBTL selects the next source task that maximizes the acquisition function (red). (CMDP: Pendulum (Time step)).

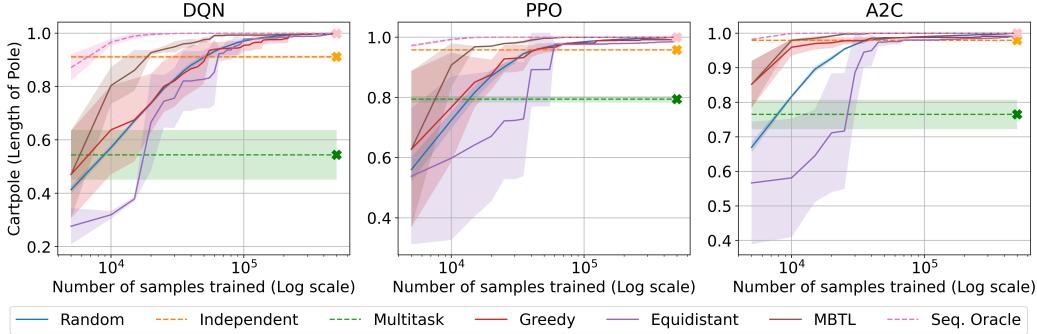


Figure 7: Sensitivity analysis on the DRL algorithm underlying MBTL (DQN, PPO, and A2C), tested on Cartpole with varying length of pole. MBTL remains effective.

domain, we considered CMDPs with varying cart mass, pole length, and pole mass. In Pendulum, we vary the timestep duration, pendulum length, and pendulum mass. The BipedalWalker was tested under varying friction, gravity, and scale. In HalfCheetah domain, we manipulated friction, gravity, and stiffness parameters. These variations critically influence the dynamics and physics of the environments. The range of context variations was selected by scaling the default values specified in CARL from 0.1 to 10 times ( $N = 100$ ), enabling a comprehensive analysis of transfer learning under drastically different conditions. We provide more details in Appendix A.4.

**Results.** The results summarized in Table 2 demonstrate MBTL’s sample efficiency and competitive performance across diverse control domains, often closely trailing the Oracle only with a small number of tasks. With the exception of the Cartpole domain, MBTL generally shows superior performance. Figure 5 shows that MBTL demonstrates rapid learning curves, reaching high normalized performance with relatively few samples. Specifically, Figure 6 illustrates the detailed process of how MBTL utilizes GP for performance estimation and chooses the next source task that maximizes the acquisition function that evaluates the expected improvement of generalized performance. MBTL achieves comparable performance to multi-task or independent baselines with 8–50x fewer samples, highlighting its effectiveness in reducing training requirements.

### 5.3.1 Sensitivity analysis

**DRL algorithms.** Figure 7 shows that MBTL remains effective with different underlying DRL algorithms—DQN, PPO, and Advantage Actor-Critic (A2C) [30]—used for single-task training.

**Acquisition functions.** Figure 8 assesses the role of acquisition functions in Bayesian optimization. While expected improvement (EI) focuses on promising marginal gains beyond the current best, UCB utilizes both mean and variance for balancing exploration and ex-

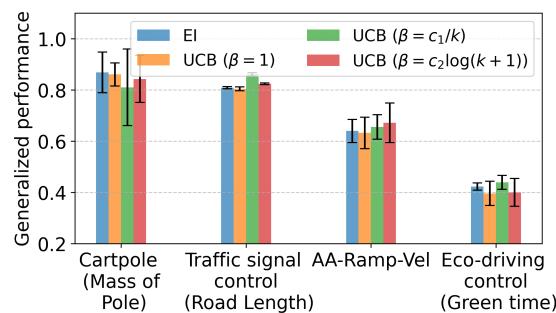


Figure 8: Sensitivity analysis on acquisition functions.

ploitation. Overall, we find that MBTL is not particularly sensitive to the choice of optimism representation in the acquisition function, which indicates that MBTL has a weak dependence on hyperparameters.

## 6 Related work

**Contextual Reinforcement Learning.** Robustness and generalization challenges in DRL are generally addressed by a few common techniques in the literature. The broader umbrella of such methods falls under CRL [6]. CRL utilizes side information about the problem variations to improve the generalization and robustness. In particular, CRL formalizes generalization in DRL using CMDPs [15, 32, 6], which incorporate context-dependent dynamics, rewards, and initial state distributions into the formalism of MDPs. This provides a structured framework to study generalization in DRL [15, 32]. The contexts of CMDPs are not always visible during training [22]. When they are visible, they can be directly used as side information by conditioning the policy on them [38].

**Multi-task learning.** A common approach for solving CRL is multi-task learning, which aims to leverage common structure between tasks. The structure between tasks has been exploited using policy sketches for task structuring [2] and by sharing a distilled policy that captures common behavior across tasks [44]. One of the known challenges in multi-task learning is that when the context features are not visible, the resulting CMDP becomes a partially observable CMDP [22, 10] and can be challenging for multi-task learning. Moreover, another challenge is *negative transfer*, where training from tasks that are too different results in training instability or failure [21, 41, 42]. In this work, we use multi-task learning as a baseline in evaluating our methods.

**Zero-shot transfer and policy reuse.** Zero-shot transfer is another commonly used technique that focuses on adapting models trained in one environment to perform effectively in unseen settings without further training [22]. Li et al. [25] introduces context-aware policy reuse, which adapts multiple source policies based on context, enhancing generalization in complex environments. Rezaei-Shoshtari et al. [34] utilize hypernetworks to approximate an RL algorithm as a mapping from parameterized CMDP to a family of near-optimal solutions to achieve zero-shot transfer across varying task conditions. Sinapov et al. [37] use meta-data to learn inter-task transferability to learn the expected benefit of transfer given a source-target task pair. Bao et al. [3] propose a metric for evaluating transferability based on information-theoretic feature representations across tasks. Another popular class of techniques for robust and generalizable DRL is domain adaptation strategies. It is often used in making learned policies adapt to new domains with varying dynamics, observations, or rewards [5]. For instance, Xing et al. [46] first learn a latent unified state representation for different domains and subsequently train DRL agents in the source domain based on that. However, our focus in this work centered around in-distribution generalization, whereas domain adaptation techniques are commonly used for out-of-distribution generalization.

**Source task selection.** In the context of transfer learning, selecting appropriate source tasks is crucial. Li and Zhang [24] proposes an optimal online method for dynamically selecting the most relevant single source policy in reinforcement learning. Beyond RL, Meiseles and Rokach [28] emphasizes structural alignment in time-series source models to prevent performance degradation, while Poth et al. [33] finds that selecting aligned intermediate tasks in natural language processing boosts transfer effectiveness. Together, these studies show the importance of context-aligned source tasks selection to improve transferability and reduce negative transfer.

## 7 Conclusion

This study introduces a method called Model-based Transfer Learning (MBTL), which layers on top of existing RL methods to effectively solve CMDPs. Rather than independent or multi-task training, which trains  $N$  or 1 models, respectively, MBTL intelligently selects an intermediate number of models to train. MBTL has two key components: an explicit model of the generalization gap and a Gaussian process component to estimate training performance. MBTL achieves up to 50x improved sample efficiency on standard and real-world benchmarks. Furthermore, MBTL achieves sublinear regret in the number of training tasks. A limitation is that MBTL is designed for a single-dimensional context variation. Promising directions of future work include studying high-dimensional context spaces and out-of-distribution generalization, as well as the development of new real-world CMDP benchmarks.

## Acknowledgments and Disclosure of Funding

The authors acknowledge the MIT SuperCloud and Lincoln Laboratory Supercomputing Center for providing HPC resources that have contributed to the research results reported within this paper. This work was supported by the National Science Foundation (NSF) CAREER award (#2239566), the Kwanjeong Educational Foundation Ph.D. scholarship program, and an Amazon Robotics Ph.D. Fellowship. The authors would like to thank the anonymous reviewers for their valuable feedback.

## References

- [1] Abubakr O Al-Abbasi, Arnob Ghosh, and Vaneet Aggarwal. Deeppool: Distributed model-free algorithm for ride-sharing using deep reinforcement learning. *IEEE Transactions on Intelligent Transportation Systems*, 20(12):4714–4727, 2019.
- [2] Jacob Andreas, Dan Klein, and Sergey Levine. Modular Multitask Reinforcement Learning with Policy Sketches. In *Proceedings of the 34th International Conference on Machine Learning*, pages 166–175. PMLR, July 2017. ISSN: 2640-3498.
- [3] Yajie Bao, Yang Li, Shao-Lun Huang, Lin Zhang, Lizhong Zheng, Amir Zamir, and Leonidas Guibas. An Information-Theoretic Approach to Transferability in Task Transfer Learning. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 2309–2313, Taipei, Taiwan, September 2019. IEEE. ISBN 978-1-5386-6249-6. doi: 10.1109/ICIP.2019.8803726.
- [4] Marc G Bellemare, Salvatore Candido, Pablo Samuel Castro, Jun Gong, Marlos C Machado, Subhodeep Moitra, Sameera S Ponda, and Ziyu Wang. Autonomous navigation of stratospheric balloons using reinforcement learning. *Nature*, 588(7836):77–82, 2020.
- [5] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013.
- [6] Carolin Benjamins, Theresa Eimer, Frederik Schubert, Aditya Mohan, Sebastian Döhler, André Biedenkapp, Bodo Rosenhahn, Frank Hutter, and Marius Lindauer. Contextualize Me – The Case for Context in Reinforcement Learning. *Transactions on Machine Learning Research*, June 2023.
- [7] Eric Brochu, Vlad M. Cora, and Nando de Freitas. A Tutorial on Bayesian Optimization of Expensive Cost Functions, with Application to Active User Modeling and Hierarchical Reinforcement Learning, December 2010.
- [8] Jung-Hoon Cho, Sirui Li, Jeongyun Kim, and Cathy Wu. Temporal transfer learning for traffic optimization with coarse-grained advisory autonomy. *arXiv preprint arXiv:2312.09436*, 2023.
- [9] Tianshu Chu, Jie Wang, Lara Codeca, and Zhaojian Li. Multi-Agent Deep Reinforcement Learning for Large-Scale Traffic Signal Control. *IEEE Transactions on Intelligent Transportation Systems*, 21(3):1086–1095, March 2020. ISSN 1524-9050, 1558-0016. doi: 10.1109/TITS.2019.2901791.
- [10] Karl Cobbe, Chris Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. In *International conference on machine learning*, pages 2048–2056. PMLR, 2020.
- [11] Jonas Degrave, Federico Felici, Jonas Buchli, Michael Neunert, Brendan Tracey, Francesco Carpanese, Timo Ewalds, Roland Hafner, Abbas Abdolmaleki, Diego de Las Casas, et al. Magnetic control of tokamak plasmas through deep reinforcement learning. *Nature*, 602(7897):414–419, 2022.
- [12] US EPA. Sources of Greenhouse Gas Emissions, 2023. URL <https://www.epa.gov/ghgemissions/sources-greenhouse-gas-emissions>.
- [13] Alhussein Fawzi, Matej Balog, Aja Huang, Thomas Hubert, Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Francisco J R Ruiz, Julian Schrittweis, Grzegorz Swirszcz, et al. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, 610(7930):47–53, 2022.
- [14] Qiangqiang Guo, Ohay Angah, Zhijun Liu, and Xuegang (Jeff) Ban. Hybrid deep reinforcement learning based eco-driving for low-level connected and automated vehicles along signalized corridors. *Transportation Research Part C: Emerging Technologies*, 124:102980, March 2021. ISSN 0968090X. doi: 10.1016/j.trc.2021.102980.

- [15] Assaf Hallak, Dotan Di Castro, and Shie Mannor. Contextual markov decision processes. *arXiv preprint arXiv:1502.02259*, 2015.
- [16] Aamir Hasan, Neeloy Chakraborty, Haonan Chen, Jung-Hoon Cho, Cathy Wu, and Katherine Driggs-Campbell. Cooperative advisory residual policies for congestion mitigation. *Journal on Autonomous Transportation Systems*, 2024.
- [17] Irina Higgins, Arka Pal, Andrei Rusu, Loic Matthey, Christopher Burgess, Alexander Pritzel, Matthew Botvinick, Charles Blundell, and Alexander Lerchner. DARLA: Improving Zero-Shot Transfer in Reinforcement Learning. In *Proceedings of the 34th International Conference on Machine Learning*, pages 1480–1490. PMLR, July 2017. ISSN: 2640-3498.
- [18] Vindula Jayawardana and Cathy Wu. Learning eco-driving strategies at signalized intersections. In *2022 European Control Conference (ECC)*, pages 383–390. IEEE, 2022.
- [19] Vindula Jayawardana, Catherine Tang, Sirui Li, Dajiang Suo, and Cathy Wu. The Impact of Task Underspecification in Evaluating Deep Reinforcement Learning. In *Advances in Neural Information Processing Systems*, volume 35, pages 23881–23893. Curran Associates, Inc., 2022.
- [20] Vindula Jayawardana, Baptiste Freydt, Ao Qu, Cameron Hickert, Edgar Sanchez, Catherine Tang, Mark Taylor, Blaine Leonard, and Cathy Wu. Mitigating metropolitan carbon emissions with dynamic eco-driving at scale. *arXiv preprint arXiv:2408.05609*, 2024.
- [21] Zhuoliang Kang, Kristen Grauman, and Fei Sha. Learning with whom to share in multi-task feature learning. In *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pages 521–528, 2011.
- [22] Robert Kirk, Amy Zhang, Edward Grefenstette, and Tim Rocktäschel. A survey of zero-shot generalisation in deep reinforcement learning. *Journal of Artificial Intelligence Research*, 76:201–264, 2023.
- [23] Li Li, Yisheng Lv, and Fei-Yue Wang. Traffic signal timing via deep reinforcement learning. *IEEE/CAA Journal of Automatica Sinica*, 3(3):247–254, July 2016. ISSN 2329-9266, 2329-9274. doi: 10.1109/JAS.2016.7508798.
- [24] Siyuan Li and Chongjie Zhang. An Optimal Online Method of Selecting Source Policies for Reinforcement Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1), April 2018. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v32i1.11718. URL <https://ojs.aaai.org/index.php/AAAI/article/view/11718>.
- [25] Siyuan Li, Fangda Gu, Guangxiang Zhu, and Chongjie Zhang. Context-aware policy reuse. *arXiv preprint arXiv:1806.03793*, 2018.
- [26] Pablo Alvarez Lopez, Michael Behrisch, Laura Bieker-Walz, Jakob Erdmann, Yun-Pang Flötteröd, Robert Hilbrich, Leonhard Lücken, Johannes Rummel, Peter Wagner, and Evamarie Wießner. Microscopic traffic simulation using sumo. In *The 21st IEEE International Conference on Intelligent Transportation Systems*. IEEE, 2018.
- [27] Daniel J Mankowitz, Andrea Michi, Anton Zhernov, Marco Gelmi, Marco Selvi, Cosmin Paduraru, Edouard Leurent, Shariq Iqbal, Jean-Baptiste Lespiau, Alex Ahern, et al. Faster sorting algorithms discovered using deep reinforcement learning. *Nature*, 618(7964):257–263, 2023.
- [28] Amiel Meiseles and Lior Rokach. Source model selection for deep learning in the time series domain. *IEEE Access*, 8:6190–6200, 2020.
- [29] Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, February 2015. ISSN 0028-0836, 1476-4687. doi: 10.1038/nature14236.
- [30] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu. Asynchronous Methods for Deep Reinforcement Learning. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1928–1937. PMLR, June 2016. ISSN: 1938-7228.
- [31] Jonas Mockus. *Bayesian Approach to Global Optimization: Theory and Applications*, volume 37 of *Mathematics and Its Applications*. Springer Netherlands, Dordrecht, 1989. ISBN 978-94-010-6898-7 978-94-009-0909-0. doi: 10.1007/978-94-009-0909-0.

- [32] Aditya Modi, Nan Jiang, Satinder Singh, and Ambuj Tewari. Markov Decision Processes with Continuous Side Information. In *Proceedings of Algorithmic Learning Theory*, pages 597–618. PMLR, April 2018. ISSN: 2640-3498.
- [33] Clifton Poth, Jonas Pfeiffer, Andreas Rücklé, and Iryna Gurevych. What to pre-train on? efficient intermediate task selection. *arXiv preprint arXiv:2104.08247*, 2021.
- [34] Sahand Rezaei-Shoshtari, Charlotte Morissette, Francois R. Hogan, Gregory Dudek, and David Meger. Hypernetworks for Zero-Shot Transfer in Reinforcement Learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(8):9579–9587, June 2023. ISSN 2374-3468, 2159-5399. doi: 10.1609/aaai.v37i8.26146.
- [35] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, August 2017.
- [36] David Silver, Aja Huang, Chris J. Maddison, Arthur Guez, Laurent Sifre, George van den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, Sander Dieleman, Dominik Grewe, John Nham, Nal Kalchbrenner, Ilya Sutskever, Timothy Lillicrap, Madeleine Leach, Koray Kavukcuoglu, Thore Graepel, and Demis Hassabis. Mastering the game of Go with deep neural networks and tree search. *Nature*, 529(7587):484–489, January 2016. ISSN 1476-4687. doi: 10.1038/nature16961.
- [37] Jivko Sinapov, Sanmit Narvekar, Matteo Leonetti, and Peter Stone. Learning Inter-Task Transferability in the Absence of Target Task Samples. In *Proceedings of the 14th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2015)*, Istanbul, Turkey, May 2015.
- [38] Shagun Sodhani, Amy Zhang, and Joelle Pineau. Multi-task reinforcement learning with context-based representations. In *International Conference on Machine Learning*, pages 9767–9779. PMLR, 2021.
- [39] Mayuri Sridhar and Cathy Wu. Piecewise Constant Policies for Human-Compatible Congestion Mitigation. In *2021 IEEE International Intelligent Transportation Systems Conference (ITSC)*, pages 2499–2505, Indianapolis, IN, USA, September 2021. IEEE. ISBN 978-1-72819-142-3. doi: 10.1109/ITSC48978.2021.9564789.
- [40] Niranjan Srinivas, Andreas Krause, Sham M. Kakade, and Matthias W. Seeger. Information-Theoretic Regret Bounds for Gaussian Process Optimization in the Bandit Setting. *IEEE Transactions on Information Theory*, 58(5):3250–3265, May 2012. ISSN 0018-9448, 1557-9654. doi: 10.1109/TIT.2011.2182033.
- [41] Trevor Standley, Amir Zamir, Dawn Chen, Leonidas Guibas, Jitendra Malik, and Silvio Savarese. Which tasks should be learned together in multi-task learning? In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 9120–9132. PMLR, 13–18 Jul 2020. URL <https://proceedings.mlr.press/v119/standley20a.html>.
- [42] Ximeng Sun, Rameswar Panda, Rogerio Feris, and Kate Saenko. Adashare: Learning what to share for efficient deep multi-task learning. *Advances in Neural Information Processing Systems*, 33:8728–8740, 2020.
- [43] Richard S. Sutton and Andrew G. Barto. *Reinforcement learning: an introduction*. Adaptive computation and machine learning series. The MIT Press, Cambridge, Massachusetts, second edition edition, 2018. ISBN 978-0-262-03924-6.
- [44] Yee Teh, Victor Bapst, Wojciech M. Czarnecki, John Quan, James Kirkpatrick, Raia Hadsell, Nicolas Heess, and Razvan Pascanu. Distral: Robust multitask reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.
- [45] Marius Wegener, Lucas Koch, Markus Eisenbarth, and Jakob Andert. Automated eco-driving in urban scenarios using deep reinforcement learning. *Transportation Research Part C: Emerging Technologies*, 126:102967, May 2021. ISSN 0968090X. doi: 10.1016/j.trc.2021.102967.
- [46] Jinwei Xing, Takashi Nagata, Kexin Chen, Xinyun Zou, Emre Neftci, and Jeffrey L Krichmar. Domain adaptation in reinforcement learning via latent unified state representation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 10452–10459, 2021.
- [47] Zhongxia Yan, Abdul Rahman Kreidieh, Eugene Vinitsky, Alexandre M. Bayen, and Cathy Wu. Unified Automatic Control of Vehicular Systems With Reinforcement Learning. *IEEE Transactions on Automation Science and Engineering*, pages 1–16, 2022. ISSN 1545-5955, 1558-3783. doi: 10.1109/TASE.2022.3168621.

## A Appendix

### Contents

|                                                               |    |
|---------------------------------------------------------------|----|
| A.1 Notation . . . . .                                        | 14 |
| A.2 Model-Based Transfer Learning (MBTL) Algorithm . . . . .  | 16 |
| A.3 Theoretical analysis . . . . .                            | 16 |
| A.3.1 Proof of Theorem 2 . . . . .                            | 16 |
| A.3.2 Proof of Corollary 2.1 . . . . .                        | 16 |
| A.3.3 Proof of Corollary 2.2 . . . . .                        | 17 |
| A.4 Experiment details . . . . .                              | 17 |
| A.4.1 Details about Gaussian process (GP) . . . . .           | 17 |
| A.4.2 Results of table with standard deviation . . . . .      | 17 |
| A.4.3 Detailed sample complexity comparison results . . . . . | 19 |
| A.4.4 Details about traffic signal control . . . . .          | 19 |
| A.4.5 Details about eco-driving control . . . . .             | 22 |
| A.4.6 Details about advisory autonomy task . . . . .          | 24 |
| A.4.7 Details of control tasks . . . . .                      | 25 |
| A.4.8 Details about Cartpole . . . . .                        | 25 |
| A.4.9 Details about Pendulum . . . . .                        | 27 |
| A.4.10 Details about BipedalWalker . . . . .                  | 28 |
| A.4.11 Details about HalfCheetah . . . . .                    | 29 |
| A.5 Potential impacts . . . . .                               | 30 |

### A.1 Notation

Table 3 describes the notation used in this paper.

Table 3: Notation used in the problem formulation

| Symbol                | Description                                                                         |
|-----------------------|-------------------------------------------------------------------------------------|
| $x$                   | Source task ( $x \in \bar{X}$ )                                                     |
| $x'$                  | Target task ( $x' \in X$ )                                                          |
| $\pi_x$               | Trained policy from source task ( $x \in \bar{X}$ )                                 |
| $x_k$                 | Selected source task at transfer step $k$ ( $k = 1, \dots, K$ )                     |
| $\mathcal{M}_x$       | Contextual MDP parameterized by $x$                                                 |
| $J(\pi_x, x)$         | Performance of task $\mathcal{M}_x$                                                 |
| $J(\pi_x, x')$        | Generalization performance (source: $x$ (or $\mathbf{x}$ ), target: $x'$ )          |
| $\Delta J(\pi_x, x')$ | Generalization gap (source: $x$ , target: $x'$ )                                    |
| $V(x'; \pi_x)$        | Expected generalization performance of source model $x$ evaluated on all $x' \in X$ |

Figure 9 helps understand the discrepancy between the observed generalized performance and the predicted one. Figure 10 illustrates how to calculate the marginal improvement of expected generalized performance ( $\hat{V}(x; x_{1:k-1}) - V(x; x_{1:k-1})$ ).

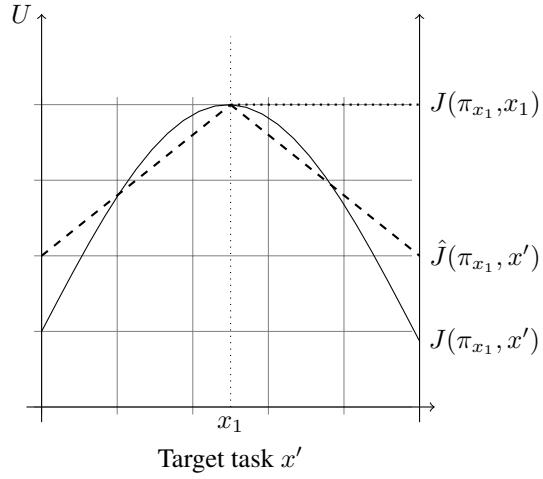


Figure 9: Illustration of the discrepancy between observed ( $J$ ) and predicted ( $\hat{J}$ ) generalized performance after training on source task  $x_1$  and attempting zero-shot transfer to  $x'$ .

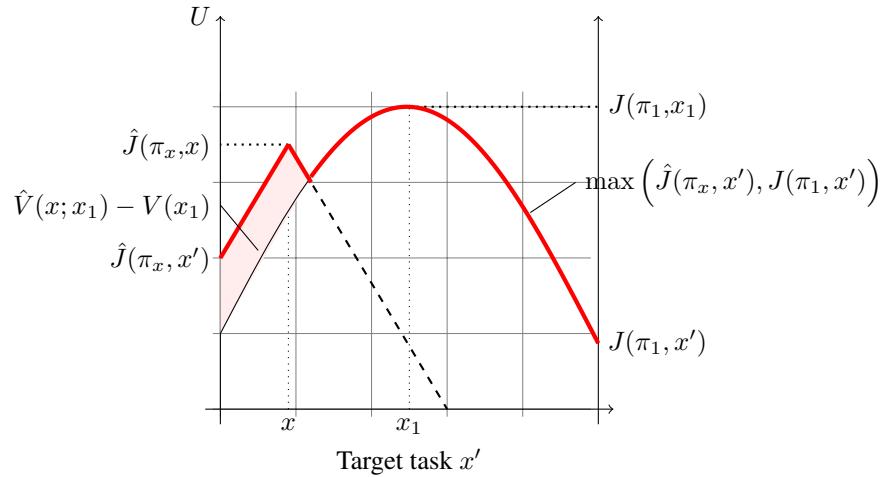


Figure 10: Step for choosing  $x_2$  that maximizes the estimated marginal improvement ( $\hat{V}(x; \pi_1) - V(x_1)$ ).  $\hat{V}(x; \pi_1)$  corresponds to the red area under the red line and  $V(x_1)$  as the area under  $J(\pi_1, x')$ .

## A.2 Model-Based Transfer Learning (MBTL) Algorithm

Model-based Transfer Learning (MBTL)

```

1: Input: CMDPs  $\mathcal{M}_x$ , Task (context) set  $X$ , Training budget  $K$ 
2: Initialize :  $J, V = 0 \forall x \in X, \pi = \{\}, k = 1$ 
3: while  $k \leq K$  do
4:   % Estimate training performance
5:    $\mu, \sigma \leftarrow \mathcal{GP}(\mathbb{E}[J(\pi_x, x)], k(x, \tilde{x}))$ 
6:   % Calculate marginal generalized performance and acquisition function
7:   Calculate  $a(x; x_{1:k-1})$  with Eq. 4
8:   % Select the next training task
9:    $x_k = \arg \max_x a(x; x_{1:k-1})$ 
10:   $\pi_k \leftarrow \text{Train}(\mathcal{M}_{x_k})$ 
11:   $\pi \leftarrow \pi \cup \{\pi_k\}$ 
12:   $k \leftarrow k + 1$ 
13: end while
14: Zero-shot transfer and calculate generalization performance  $V(x_1, \dots, x_k)$ 
15: Output: Set of policies  $\pi$  and generalization performance  $V$ 
```

## A.3 Theoretical analysis

### A.3.1 Proof of Theorem 2

**Theorem 2.** For a given  $\delta' \in (0, 1)$  and scaling factor  $\beta_k = 2 \log(|X| \pi^2 k^2 / 6\delta)$ , the cumulative regret  $R_K$  is bounded by  $\sqrt{C_1 \beta_K \gamma_K \sum_{k=1}^K \left( \frac{|X_k|}{|X|} \right)^2}$  with probability at least  $1 - \delta'$ .

*Proof.* The following Lemma 3 and 4 is basically considering the cardinality of restricted search space of  $X_k$  instead of  $X$  upon the lemmas in literature [40].

**Lemma 3.** For  $t \geq 1$ , if  $|f(x) - \mu_{k-1}(x)| \leq \beta_k^{1/2} \sigma_{k-1}(x) \quad \forall x \in X_k$ , then the regret  $r_t$  is bounded by  $2|X_k| \beta_k^{1/2} \sigma_{k-1}(x) / |X|$ .

**Lemma 4.** Setting  $\delta \in (0, 1)$ ,  $\beta_k = 2 \log(|X| \pi^2 k^2 / 6\delta)$ , and  $C_1 := \frac{8}{\log(1+\sigma^{-2})} \geq 8\sigma^2$ , we have  $\Pr \left[ \sum_{k=1}^K r_k \left( \frac{|X|}{|X_k|} \right)^2 \leq C_1 \beta_K \gamma_K \quad \forall K \geq 1 \right] \geq 1 - \delta$ .

Using Lemma 3, Lemma 5.3 in [40], Lemma 4, and Cauchy–Schwarz inequality, we can bound the cumulative regret as:

$$R_K = \sum_{k=1}^K r_k \leq \sqrt{\sum_{k=1}^K r_k \left( \frac{|X|}{|X_k|} \right)^2 \sum_{k=1}^K \left( \frac{|X_k|}{|X|} \right)^2} \leq \sqrt{C_1 \beta_K \gamma_K \sum_{k=1}^K \left( \frac{|X_k|}{|X|} \right)^2}. \quad (5)$$

□

### A.3.2 Proof of Corollary 2.1

**Corollary 2.1.** Consider  $|X_k| = \frac{1}{\sqrt{k}} |X|$ . The regret bound would be  $R_K \leq \sqrt{C_1 \beta_K \gamma_K \log K}$  with a probability of at least  $1 - \delta'$ .

*Proof.* Recall that  $\sum_{k=1}^K \frac{1}{k} \leq \log K$ .

Calculating the sum of squares for the reduced segments, we have:

$$\sum_{k=1}^K |X_k|^2 = \sum_{k=1}^K \frac{1}{k} |X|^2 \leq |X|^2 \log K \quad (6)$$

The cumulative regret can be bounded as below:

$$R_K = \sum_{k=1}^K r_k \leq \sqrt{C_1 \beta_K \gamma_K \sum_{k=1}^K \left( \frac{|X_k|}{|X|} \right)^2} \leq \sqrt{C_1 \beta_K \gamma_K \log K}. \quad (7)$$

□

### A.3.3 Proof of Corollary 2.2

**Corollary 2.2.** *The regret bound for the  $|X_k| \leq 2^{-\lfloor \log_2 k \rfloor} |X|$  would be  $R_K \leq \sqrt{C_1 \beta_K \gamma_K \pi^2 / 6}$  with a probability of at least  $1 - \delta'$ .*

*Proof.* Calculating the sum of squares for the reduced segments, we have:

$$\sum_{k=1}^K |X_k|^2 = \sum_{k=1}^K 2^{-2\lfloor \log_2 k \rfloor} |X|^2 \leq \frac{1}{k^2} |X|^2 \leq \frac{\pi^2}{6} |X|^2 \quad (8)$$

The cumulative regret can be bounded as below:

$$R_K = \sum_{k=1}^K r_k \leq \sqrt{C_1 \beta_K \gamma_K \sum_{k=1}^K \left( \frac{|X_k|}{|X|} \right)^2} \leq \sqrt{\frac{C_1 \beta_K \gamma_K \pi^2}{6}}. \quad (9)$$

□

## A.4 Experiment details

### A.4.1 Details about Gaussian process (GP)

In our study, we conducted hyperparameter tuning experiments for Gaussian Process (GP) regression to optimize its performance. Specifically, we varied the noise standard deviation over the set  $\{0.001, 0.01, 0.1, 1\}$  and the number of restarts for the optimizer over the set  $\{5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15\}$  with several kernel configurations. We set noise standard deviation as 0.1, the number of restarts for the optimizer over the set as 9, and the kernel as the combination of a constant kernel (C) with the bound from 0.001 to 1000 and a radial basis function kernel (RBF) with a length scale ranging from 0.01 to 100. The noise level was set to the square of the noise standard deviation ( $0.1^2$ ).

### A.4.2 Results of table with standard deviation

Table 4: Comparative performance of different methods on context-variant traffic and benchmark CMDPs

| Benchmark (CMDP)         |                   | Baselines                 |                           | Sequential Baselines      |                           | MBTL                      | Oracle                    |                    |
|--------------------------|-------------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|---------------------------|--------------------|
| Domain                   | Context Variation | Independent               | Multi-task                | Random                    | Equidistant               | Greedy                    | Ours                      | Sequential         |
| Number of Trained Models |                   | <i>N</i>                  | 1                         | <i>k</i>                  | <i>K</i>                  | <i>k</i>                  | <i>k</i>                  | <i>N</i>           |
| <b>Traffic Signal</b>    | Road Length       | <b>0.9409</b><br>(0.0002) | 0.8242<br>(0.0538)        | 0.9249<br>(0.0007)        | 0.9213<br>(0.0001)        | 0.9278<br>(0.0071)        | 0.9371<br>(0.0023)        | 0.9409<br>(0.0006) |
|                          | Inflow            | 0.8646<br>(0.0009)        | 0.8319<br>(0.0040)        | 0.8457<br>(0.0028)        | <b>0.8700</b><br>(0.0051) | 0.8496<br>(0.0039)        | <b>0.8673</b><br>(0.0016) | 0.8768<br>(0.0012) |
| <b>Traffic Signal</b>    | Speed Limit       | 0.8857<br>(0.0005)        | 0.6083<br>(0.0403)        | <b>0.8821</b><br>(0.0003) | <b>0.8858</b><br>(0.0004) | <b>0.8862</b><br>(0.0010) | <b>0.8854</b><br>(0.0008) | 0.8876<br>(0.0002) |
| <b>Eco-Driving</b>       | Penetration Rate  | 0.5260<br>(0.0087)        | 0.1945<br>(0.0057)        | 0.5959<br>(0.0058)        | 0.5934<br>(0.0052)        | 0.5827<br>(0.0151)        | <b>0.6323</b><br>(0.0235) | 0.6660<br>(0.0049) |
|                          | Inflow            | 0.4061<br>(0.0094)        | 0.2229<br>(0.0010)        | 0.4774<br>(0.0097)        | 0.4705<br>(0.0051)        | 0.4673<br>(0.0124)        | <b>0.5108</b><br>(0.0175) | 0.5528<br>(0.0096) |
| <b>Eco-Driving</b>       | Green Phase       | 0.3850<br>(0.0063)        | 0.4228<br>(0.0184)        | 0.4406<br>(0.0045)        | <b>0.4557</b><br>(0.0334) | 0.4431<br>(0.0109)        | <b>0.4700</b><br>(0.0144) | 0.5027<br>(0.0058) |
| <b>AA-Ring-Acc</b>       | Hold Duration     | 0.8362<br>(0.0048)        | <b>0.9209</b><br>(0.0235) | 0.8924<br>(0.0097)        | <b>0.9057</b><br>(0.0338) | 0.8776<br>(0.0103)        | <b>0.9242</b><br>(0.0210) | 0.9552<br>(0.0132) |
|                          | Vel               | 0.9589<br>(0.0096)        | 0.9720<br>(0.0035)        | 0.9785<br>(0.0012)        | 0.9772<br>(0.0072)        | 0.9807<br>(0.0007)        | <b>0.9816</b><br>(0.0004) | 0.9822<br>(0.0000) |
| <b>AA-Ramp-Acc</b>       | Hold Duration     | 0.4276<br>(0.0066)        | 0.5158<br>(0.045)         | <b>0.6050</b><br>(0.0192) | 0.5956<br>(0.1052)        | <b>0.6143</b><br>(0.0793) | <b>0.6318</b><br>(0.0272) | 0.7111<br>(0.0479) |
|                          | Vel               | 0.5473<br>(0.0222)        | 0.5034<br>(0.014)         | 0.6690<br>(0.0476)        | <b>0.6787</b><br>(0.0544) | 0.5907<br>(0.0407)        | <b>0.7182</b><br>(0.0461) | 0.7686<br>(0.0581) |
| <b>Pendulum</b>          | Length            | 0.7383<br>(0.0034)        | 0.6830<br>(0.0008)        | 0.7270<br>(0.0008)        | 0.7092<br>(0.0027)        | 0.7327<br>(0.005)         | <b>0.7697</b><br>(0.0130) | 0.7969<br>(0.0123) |
|                          | Mass              | 0.6237<br>(0.0023)        | 0.5793<br>(0.0041)        | 0.6329<br>(0.0077)        | 0.6092<br>(0.0013)        | 0.6408<br>(0.0073)        | <b>0.6827</b><br>(0.0239) | 0.7132<br>(0.0129) |
| <b>Pendulum</b>          | Timestep          | 0.8135<br>(0.0103)        | 0.7247<br>(0.0488)        | 0.7989<br>(0.0073)        | 0.7541<br>(0.0231)        | <b>0.8177</b><br>(0.0024) | 0.8141<br>(0.0032)        | 0.8801<br>(0.0198) |
| <b>Cartpole</b>          | Mass of Cart      | <b>0.9466</b><br>(0.0065) | 0.7153<br>(0.2195)        | 0.7221<br>(0.0188)        | 0.7516<br>(0.0877)        | 0.6501<br>(0.0268)        | 0.8212<br>(0.0164)        | 0.9838<br>(0.0131) |
|                          | Length of Pole    | <b>0.9110</b><br>(0.0065) | 0.5441<br>(0.1614)        | 0.8121<br>(0.0156)        | 0.8428<br>(0.1167)        | 0.8217<br>(0.0980)        | <b>0.9124</b><br>(0.0266) | 0.9875<br>(0.0047) |
| <b>Cartpole</b>          | Mass of Pole      | <b>0.9560</b><br>(0.0128) | 0.6073<br>(0.0948)        | 0.8858<br>(0.0052)        | 0.7909<br>(0.0548)        | 0.8744<br>(0.0804)        | 0.9351<br>(0.0403)        | 1.0000<br>(0.0000) |
| <b>BipedalWalker</b>     | Gravity           | 0.9281<br>(0.0034)        | 0.7898<br>(0.0928)        | 0.9330<br>(0.0025)        | <b>0.9494</b><br>(0.0119) | 0.9359<br>(0.0172)        | <b>0.9393</b><br>(0.0123) | 0.9674<br>(0.0014) |
|                          | Friction          | 0.9317<br>(0.0074)        | 0.9051<br>(0.0734)        | 0.9650<br>(0.0021)        | 0.9664<br>(0.004)         | 0.9645<br>(0.0098)        | <b>0.9713</b><br>(0.0008) | 0.9778<br>(0.0013) |
| <b>BipedalWalker</b>     | Scale             | 0.8694<br>(0.0087)        | 0.7452<br>(0.0938)        | 0.8605<br>(0.0081)        | 0.8496<br>(0.0141)        | <b>0.8792</b><br>(0.0186) | <b>0.8886</b><br>(0.0183) | 0.9107<br>(0.004)  |
| <b>HalfCheetah</b>       | Gravity           | 0.6679<br>(0.0162)        | 0.6292<br>(0.0258)        | 0.8542<br>(0.015)         | 0.8634<br>(0.0493)        | 0.8663<br>(0.0563)        | <b>0.9073</b><br>(0.0133) | 0.9544<br>(0.0221) |
|                          | Friction          | 0.6693<br>(0.0203)        | 0.7242<br>(0.1056)        | 0.8567<br>(0.036)         | 0.8703<br>(0.033)         | 0.8591<br>(0.0397)        | <b>0.9274</b><br>(0.0392) | 0.9663<br>(0.0276) |
| <b>HalfCheetah</b>       | Stiffness         | 0.6561<br>(0.0101)        | 0.7007<br>(0.1126)        | 0.8533<br>(0.0163)        | 0.7817<br>(0.0419)        | 0.8785<br>(0.0138)        | <b>0.9146</b><br>(0.0189) | 0.9674<br>(0.0286) |
| <b>Average</b>           |                   | 0.7775<br>(0.197)         | 0.6337<br>(0.2395)        | 0.8068<br>(0.1551)        | 0.7996<br>(0.1548)        | 0.8037<br>(0.1629)        | 0.8435<br>(0.1501)        | 0.8833<br>(0.1467) |

\* Note: Bold values represent the highest value(s) within the statistically significant range for each task, excluding the oracle. Standard deviation across multiple runs in the parenthesis.

‡AA: Advisory autonomy tasks, Ring: Single lane ring, Ramp: Highway ramp, Acc: Acceleration guidance, Vel: Speed guidance.

#### A.4.3 Detailed sample complexity comparison results

Table 5 presents a comparison of sample complexity required for MBTL to perform as good as the best generalization performance of baselines (Independent training and multi-task training) across various tasks in the CMDP. Each row lists a different domain, the specific context variation applied (e.g., changes in physical properties or environmental parameters), and two key values:  $k^*$  and  $N$ , where  $k^*$  represents the number of models required by MBTL to reach a performance level comparable to the baseline. This value is shown as a range (e.g., [3, 5, 3]), indicating results from three random seeds.  $N$  represents the total number of contexts. The value  $\frac{N}{k^*}$  helps illustrate the sample efficiency of MBTL.

Table 5: Sample complexity comparison to baseline performance on CMDP tasks

| Task           | Context Variation | $k^*$      | $N$ |
|----------------|-------------------|------------|-----|
| Pendulum       | Length            | [3, 5, 3]  | 100 |
| Pendulum       | Mass              | [3, 3, 3]  | 100 |
| Pendulum       | Timestep          | [6, 5, 6]  | 100 |
| Cartpole       | Mass of Cart      | [9, 13, 6] | 100 |
| Cartpole       | Length of Pole    | [4, 4, 3]  | 100 |
| Cartpole       | Mass of Pole      | [3, 3, 3]  | 100 |
| BipedalWalker  | Gravity           | [2, 2, 3]  | 100 |
| BipedalWalker  | Friction          | [2, 2, 2]  | 100 |
| BipedalWalker  | Scale             | [3, 3, 1]  | 100 |
| HalfCheetah    | Gravity           | [2, 2, 2]  | 100 |
| HalfCheetah    | Friction          | [1, 2, 2]  | 100 |
| HalfCheetah    | Stiffness         | [1, 2, 2]  | 100 |
| AA-Ring-Acc    | Hold Duration     | [3, 8, 2]  | 40  |
| AA-Ring-Vel    | Hold Duration     | [2, 1, 1]  | 40  |
| AA-Ramp-Acc    | Hold Duration     | [1, 2, 2]  | 40  |
| AA-Ramp-Vel    | Hold Duration     | [2, 2, 2]  | 40  |
| Traffic Signal | Road Length       | [8, 5, 6]  | 50  |
| Traffic Signal | Inflow            | [2, 2, 2]  | 50  |
| Traffic Signal | Speed Limit       | [4, 4, 5]  | 50  |
| Eco-Driving    | Penetration Rate  | [2, 2, 1]  | 50  |
| Eco-Driving    | Inflow            | [2, 1, 1]  | 50  |
| Eco-Driving    | Green Phase       | [2, 2, 2]  | 50  |

#### A.4.4 Details about traffic signal control

Most traffic lights operate on fixed schedules, but adaptive traffic signal control using DRL can optimize the traffic flow using real-time information on the traffic [9, 23], though challenges persist in generalizing across various intersection configurations [19].

Figure 11 showcases the layout of traffic networks used in a traffic signal control task with several lanes and a signalized intersection in the middle. The state space represents the presence of vehicles in discretized lane cells along the incoming roads. Actions determine which lane gets the green phase of the traffic signal, and rewards are based on changes in cumulative stopped time, the period when speed is zero. The global objective is to minimize the average waiting times at the intersection. Different configurations of intersections (e.g., road length, inflow, speed limits) are modeled to represent varying real-world conditions.

**Training configuration** We used the microscopic traffic simulation called Simulation of Urban MObility (SUMO) [26] v.1.16.0 and PPO for RL algorithm [35]. All experiments are done on a distributed computing cluster equipped with 48 Intel Xeon Platinum 8260 CPUs.

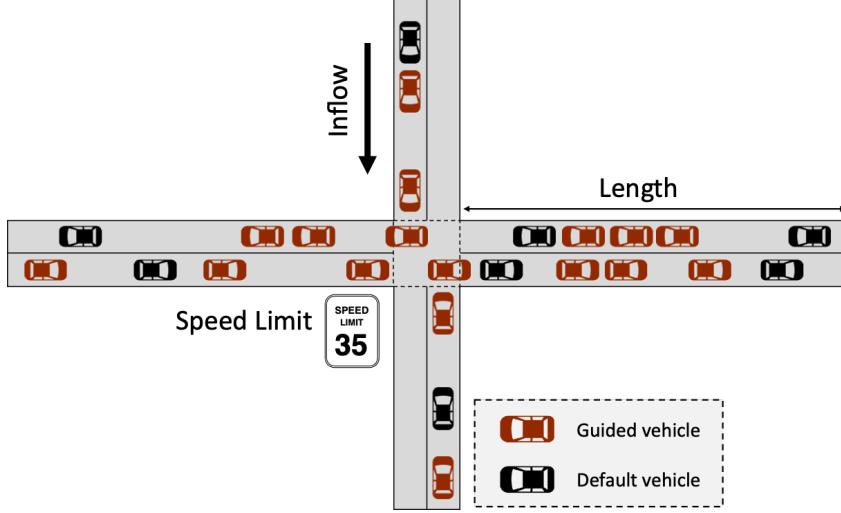


Figure 11: Illustration of the traffic networks in traffic signal control task.

**Transferability heatmap** Figure 12 presents heatmaps of transferability for different traffic signal control tasks, each varying a specific aspect: inflow, speed limit, and road length. The heatmaps display the effectiveness of strategy transfer from each source task (vertical axis) to each target task (horizontal axis). In terms of inflow variation, transferability drops when transferring from tasks with lower vehicle inflow to those with higher inflow. In speed limit variation, the transferability shows uniform effectiveness, suggesting less sensitivity to these changes. In road length variation, distinct blocks of high transferability indicate that different road lengths may require significantly tailored strategies.

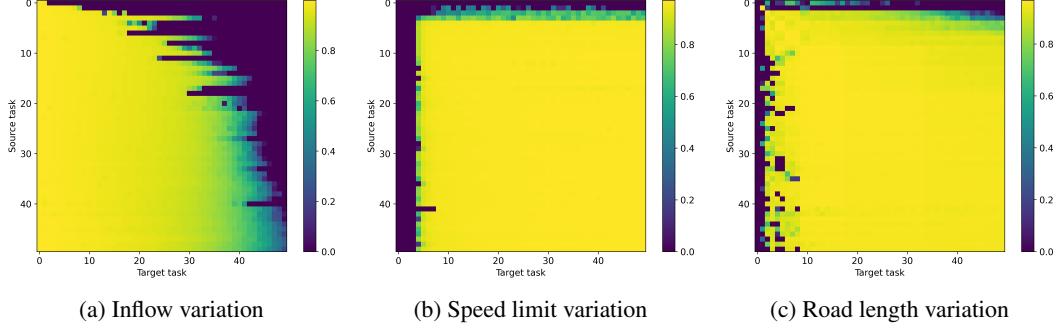


Figure 12: Examples of transferability heatmap for traffic signal control.

**Results** Figure 13 illustrates the normalized generalized performance across various traffic control tasks: inflow, speed limit, and road length. The plots display how different strategies adapt with increasing transfer steps:

- **Inflow:** Performance improves as the number of transfer steps increases, with MBTL strategy consistently achieving the highest scores, demonstrating their effectiveness in adapting to changes in inflow conditions.
- **Speed Limit:** Here, performance levels are relatively stable across all strategies except for the multitask training.
- **Road Length:** There is a general upward trend in performance for all strategies, particularly for MBTL, indicating robustness in adapting to different road lengths.

This data suggests that MBTL and Oracle are particularly effective across varying conditions, maintaining higher levels of performance adaptability.

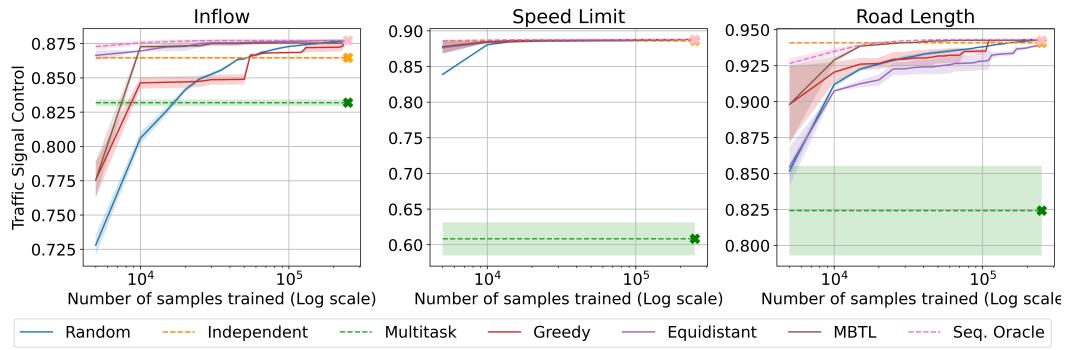


Figure 13: Comparison of normalized generalized performance of all target tasks: Traffic signal control.

#### A.4.5 Details about eco-driving control

Given the significant portion of greenhouse gas emissions in the United States coming from the transportation sector [12], eco-driving behaviors are critical for climate change mitigation. Deep reinforcement learning-based eco-driving strategies have been developed [14, 45, 18, 20] but still have some issues of difficulties in generalization. We also extend to various intersection configurations with different traffic inflow rates, penetration rates of eco-driving systems, and durations of green phases at static traffic signals to optimize vehicle behaviors for reduced emissions.

Figure 14 illustrates the traffic road network used in the eco-driving control task. The road network is depicted as traffic flowing vertically and horizontally, crossing the static phase traffic signal. There are both guided and default vehicles in the system. The state space includes the speed and position of the ego vehicle, the leading vehicle, and the following vehicles, supplemented by the current traffic signal phase and relevant context features, including lane length and green phase durations. The action space specifically focuses on the ego vehicle’s acceleration control. The reward mechanism is designed to optimize the driving strategy by balancing the average speed of the vehicles against penalties for emissions, thereby promoting eco-friendly driving behaviors within the traffic system.

**Training configuration** We also used the microscopic traffic simulation called Simulation of Urban MObility (SUMO) [26] v.1.16.0 and PPO for RL algorithm [35].

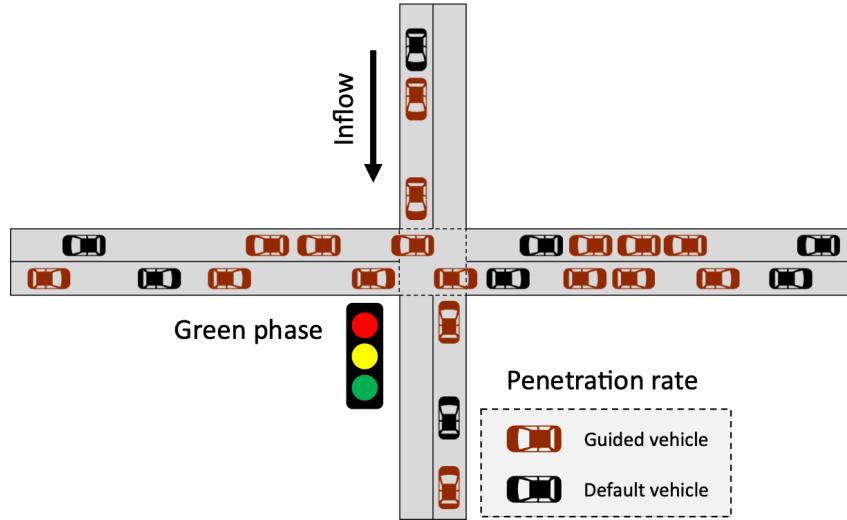


Figure 14: Illustration of the traffic networks in eco-driving control task.

**Transferability heatmap** Figure 15 displays heatmaps for the eco-driving control task, with each heatmap varying an aspect such as green phase, inflow, and penetration rate. These visuals illustrate the transferability of strategies from source tasks (vertical axis) to target tasks (horizontal axis), highlighting the impact of traffic light phases, vehicle inflow, and the proportion of guided vehicles on strategy effectiveness. Notably, longer green phases correlate with improved performance and transferability. For inflow variations, reduced inflow typically yields better outcomes. However, variations in the penetration rate of guided vehicles show minimal impact on performance differences.

**Results** Figure 16 illustrates the normalized generalized performance across variants of eco-driving control tasks, specifically looking at variations in green phase time, inflow, and penetration rate. The graphs depict performance enhancement over transfer steps for different strategies. Notably, MBTL consistently demonstrates superior performance across all variations, indicating robust adaptability to changing task parameters.

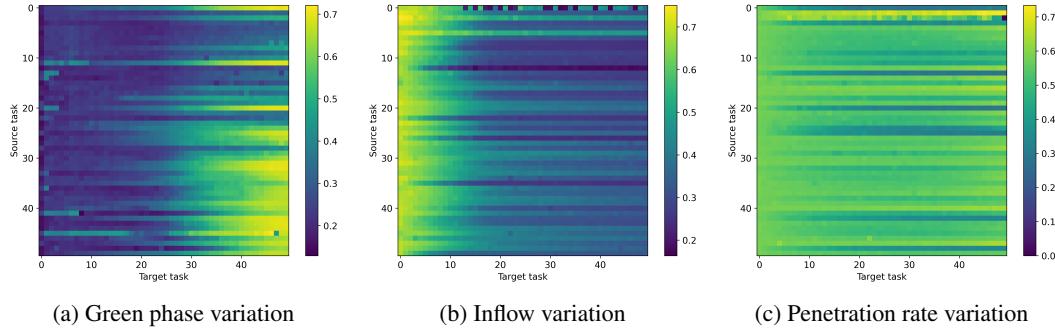


Figure 15: Examples of transferability heatmap for eco-driving control.

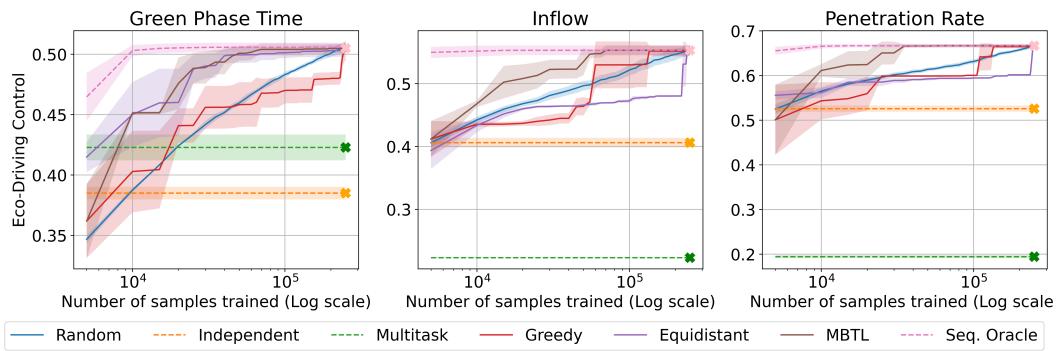


Figure 16: Comparison of normalized generalized performance of all target tasks: Eco-driving control.

#### A.4.6 Details about advisory autonomy task

Advisory autonomy involves a real-time speed advisory system that enables human drivers to emulate the system-level performance of autonomous vehicles in mixed autonomy systems [39, 8, 16]. Instead of direct and instantaneous control, human drivers receive periodic guidance, which varies based on road type and guidance strategy. Here, we consider the different frequencies of this periodic guidance as contextual MDPs since the zero-order hold action affects the transition function.

Figure 17 illustrates two distinct traffic network configurations used in the advisory autonomy task: a single-lane ring and a highway ramp. The single-lane ring features 22 vehicles circulating the ring, with only one being actively controlled, presenting a relatively controlled environment for testing vehicle guidance systems. The highway ramp scenario introduces a more complex dynamic, where vehicles not only travel along the highway but also merge from ramps, creating potential stop-and-go traffic patterns that challenge the adaptability of autonomous guidance systems.

**Problem Definition:** In a single-lane scenario, the state space includes the speeds of the ego and leading vehicles, along with the headway. For highway ramp scenarios, additional states cover the relative positions and speeds of adjacent vehicles. Actions vary by guidance type: for acceleration guidance, the action space is continuous, ranging from  $-1$  to  $1$ ; for speed guidance, it has ten discrete actions compared to the speed limit. Rewards are based on system throughput or average speed of all vehicles in the system.

**Context Variations:** We explore different durations of coarse-grained guidance holds to test various levels of human compatibility, adjusting the model based on observed driver behaviors and system performance.

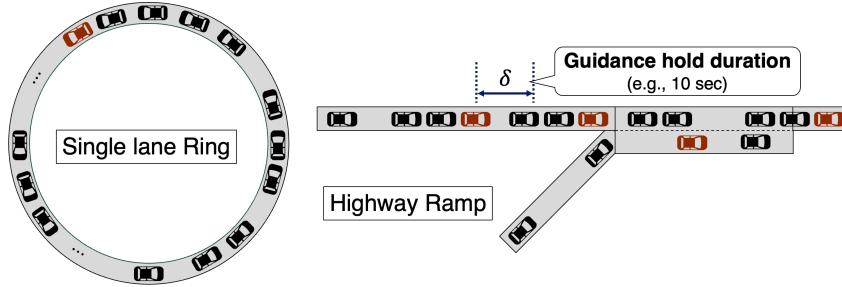


Figure 17: Illustration of the traffic networks in advisory autonomy task.

**Transferability heatmap** Figure 18 showcases heatmaps of transferability for advisory autonomy tasks, each varying in specific aspects: acceleration guidance and speed guidance across a single lane ring and a highway ramp. These heatmaps demonstrate the effectiveness of strategy transfer from each source task (vertical axis) to each target task (horizontal axis), capturing how variations in task conditions influence adaptability. For acceleration guidance in a ring setup (a), transferability is generally higher among tasks with similar acceleration demands. In contrast, speed guidance on a ramp (d) reveals more variability in transferability, potentially due to the complexity of speed adjustments in ramp scenarios.

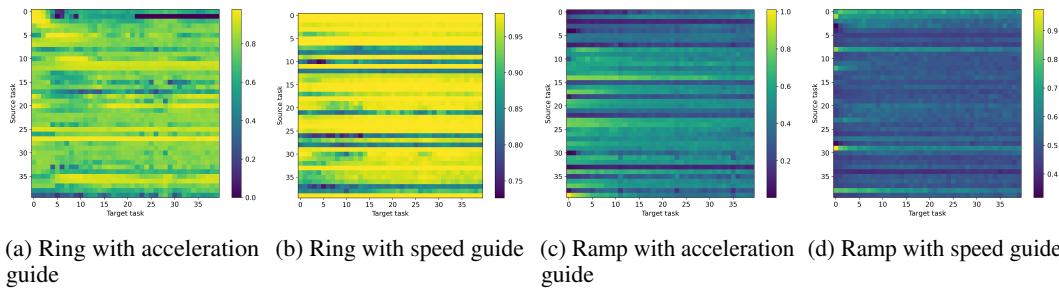


Figure 18: Examples of transferability heatmap for advisory autonomy.

**Results** Figure 19 illustrates the comparison of normalized generalized performance for advisory autonomy tasks, specifically acceleration and speed guidance in a ring and acceleration guidance on a ramp. The graphs demonstrate that MBTL consistently exhibits higher performance across all tasks. Particularly, acceleration guidance in both ring and ramp scenarios shows significant performance improvements over transfer steps, with MBTL closely matching in some instances.

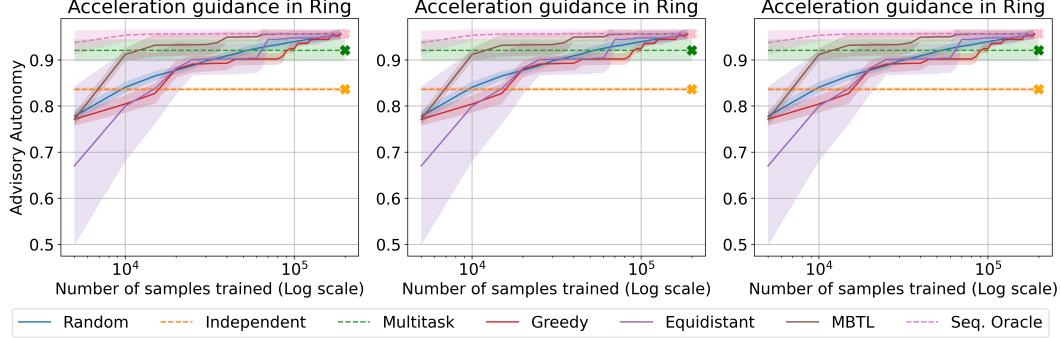


Figure 19: Comparison of normalized generalized performance of all target tasks: Advisory autonomy.

#### A.4.7 Details of control tasks

For this experimental phase, we selected context-extended versions of standard RL environments from the CARL benchmark library, including Cartpole, Pendulum, BipedalWalker, and Halfcheetah. These environments were chosen to rigorously test the robustness and adaptability of our MBTL algorithm under varied conditions that mirror the complexity encountered in real-world scenarios.

**Context Variations:** In the Cartpole tasks, we explored CMDPs with varying cart masses, pole lengths, and pole masses. For the Pendulum, the experiments involved adjusting the timestep duration, pendulum length, and pendulum mass. The BipedalWalker was tested under different settings of friction, gravity, and scale. Similarly, in the Halfcheetah tasks, we manipulated parameters such as friction, gravity, and stiffness to simulate different physical conditions. These variations critically influence the dynamics and physics of the environments, thereby presenting unique challenges that test the algorithm’s capacity to generalize from previous learning experiences without the need for extensive retraining. The range of context variations was established by scaling the default values specified in the CARL framework from 0.1 to 10 times, enabling a comprehensive examination of each model’s performance under drastically different conditions.

**License:** CARL falls under the Apache License 2.0 as is permitted by all work that we use [6].

#### A.4.8 Details about Cartpole

**Transferability heatmap** Figure 20 presents transferability heatmaps for the Cartpole task with variations in three physical properties: mass of the cart, length of the pole, and mass of the pole. Each heatmap illustrates how well strategies transfer from source tasks (vertical axis) to target tasks (horizontal axis), depicting the influence of each parameter on control strategy effectiveness. For the mass of the cart variation (a), transferability decreases as the mass difference increases. In the length of the pole variation (b), strategies are less transferable between significantly different pole lengths. Similarly, for the mass of the pole variation (c), variations show divergent transferability depending on the extent of mass change.

**Results** Figure 21 presents a comparison of normalized generalized performance for the Cartpole task across different strategies when varying the mass of the cart, length of the pole, and mass of the pole.

In the mass of cart variation, the performance generally increases with transfer steps, with MBTL strategies achieving the highest scores, indicating robust adaptability to changes in cart mass. Similar trends are observed with length variation and mass of pole variation. MBTL shows close to oracle performance.

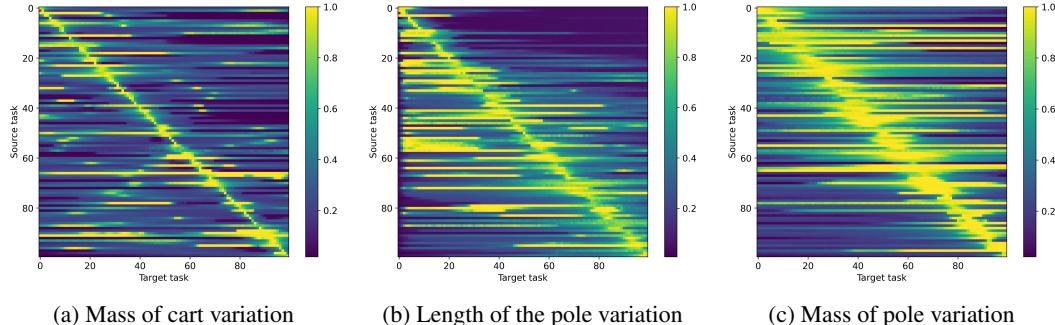


Figure 20: Examples of transferability heatmap for Cartpole.

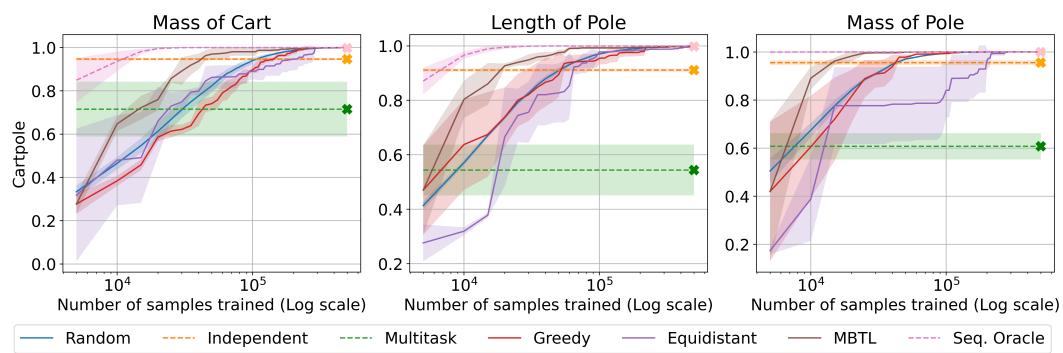


Figure 21: Comparison of normalized generalized performance of all target tasks: Cartpole.

#### A.4.9 Details about Pendulum

**Transferability heatmap** Figure 22 presents transferability heatmaps for the Pendulum task with variations in three physical properties: timestep, length of the pendulum, and mass of the pendulum. Each heatmap illustrates how effectively strategies transfer from source tasks (vertical axis) to target tasks (horizontal axis), highlighting the impact of each parameter on control strategy effectiveness. For the timestep variation (a), there appears to be high consistency in transferability across different timesteps, especially around the diagonal axis. In the length of the pendulum variation (b), transferability decreases with greater length differences. Similarly, for the mass of the pendulum variation (c), transferability shows variability dependent on the extent of mass changes.

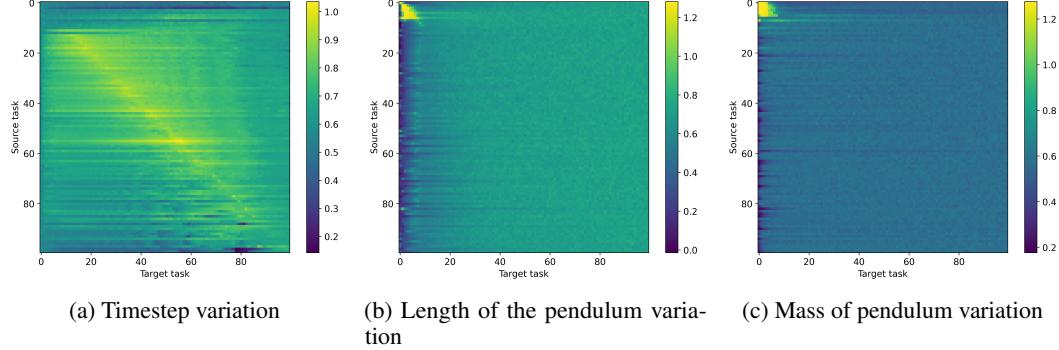


Figure 22: Examples of transferability heatmap for Pendulum.

**Results** Figure 23 shows a comparison of normalized generalized performance for the Pendulum task across different strategies when varying the timestep, length of the pendulum, and mass of the pendulum. For the length of the pendulum variation and mass of the pendulum one, MBTL strategies demonstrate the highest scores, suggesting robust adaptability to changes in pendulum dynamics. MBTL shows performance close to that of the Oracle across all variations, indicating its effectiveness in handling dynamic changes in system parameters.

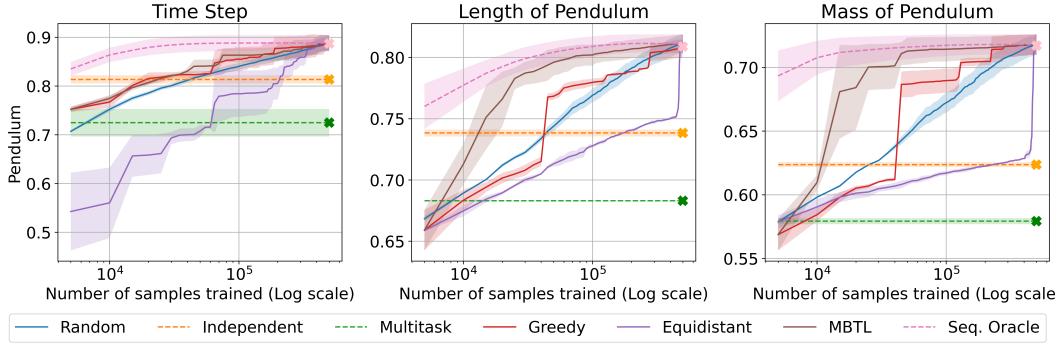


Figure 23: Comparison of normalized generalized performance of all target tasks: Pendulum.

#### A.4.10 Details about BipedalWalker

**Transferability heatmap** Figure 24 presents transferability heatmaps for the BipedalWalker task, focusing on three variations: friction, gravity, and scale. Each heatmap illustrates the effectiveness of strategy transfer from source tasks (vertical axis) to target tasks (horizontal axis), highlighting how each parameter influences control strategy adaptability. For friction variation (a), strategies show uniform transferability across different friction levels. In gravity variation (b), transferability is highly variable, suggesting that strategies need specific tuning for different gravity levels. For scale variation (c), the heatmap indicates variable transferability, reflecting the challenges of scaling control strategies.

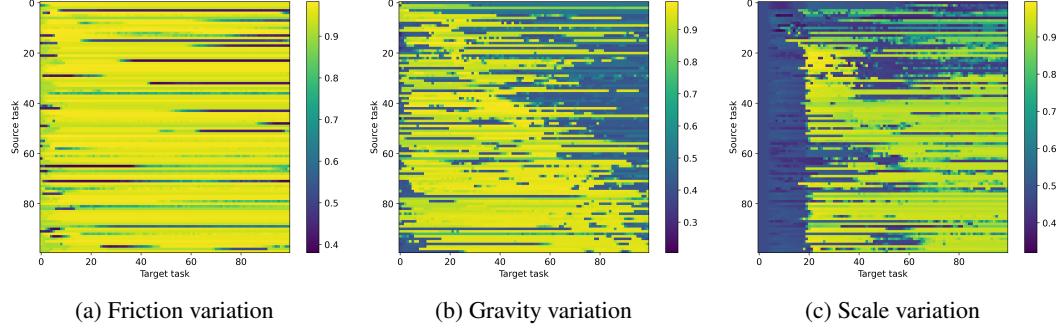


Figure 24: Examples of transferability heatmap for BipedalWalker.

**Results** Figure 25 shows the comparison of normalized generalized performance for all variations within the BipedalWalker task. There is no huge difference in performance for all three cases, but if we look into the tabular results in Table 2, MBTL shows the highest performance across varying conditions, indicating their robustness in adapting to changes in physical parameters of the model. This suggests that these strategies are more effective in handling the complexities introduced by different frictions, gravities, and scales compared to other baselines.

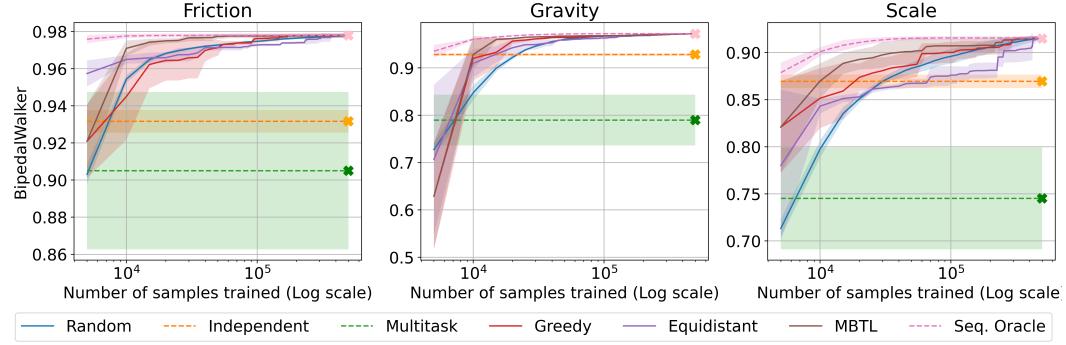


Figure 25: Comparison of normalized generalized performance of all target tasks: BipedalWalker.

#### A.4.11 Details about HalfCheetah

**Transferability heatmap** Figure 26 displays transferability heatmaps for the HalfCheetah task, focusing on three physical properties: friction, gravity, and stiffness. Each heatmap demonstrates the transferability of strategies from source tasks (vertical axis) to target tasks (horizontal axis). For friction variation (a), there is uniform high transferability across different friction levels, indicating that strategies are robust to changes in friction. Gravity variation (b) shows less consistent transferability, suggesting a sensitivity to gravity changes that might require adaptation of strategies. Stiffness variation (c) similarly demonstrates variable transferability, highlighting the challenges of adapting to different stiffness levels in control strategies.

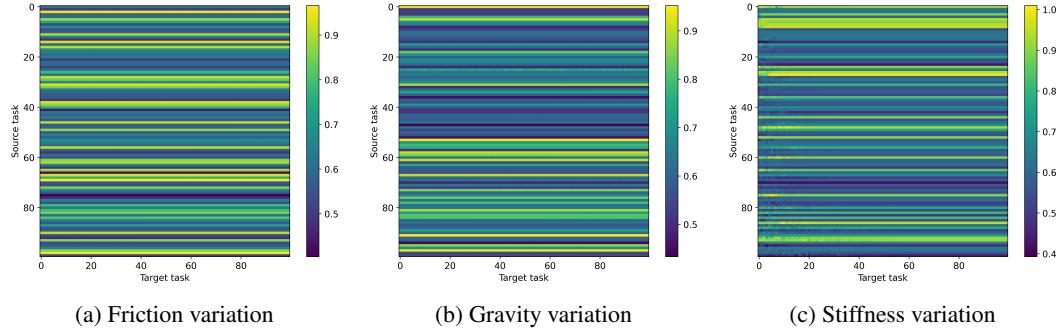


Figure 26: Examples of transferability heatmap for HalfCheetah.

**Results** Figure 27 presents a comparison of normalized generalized performance across various strategies for the HalfCheetah task with respect to the varied physical properties. The results indicate that the MBTL generally outperforms others, particularly in managing variations in gravity and stiffness, suggesting the superior adaptability of these models to physical changes in the task environment. The trends across different parameters confirm the critical impact of task-specific dynamics on the effectiveness of the tested strategies.

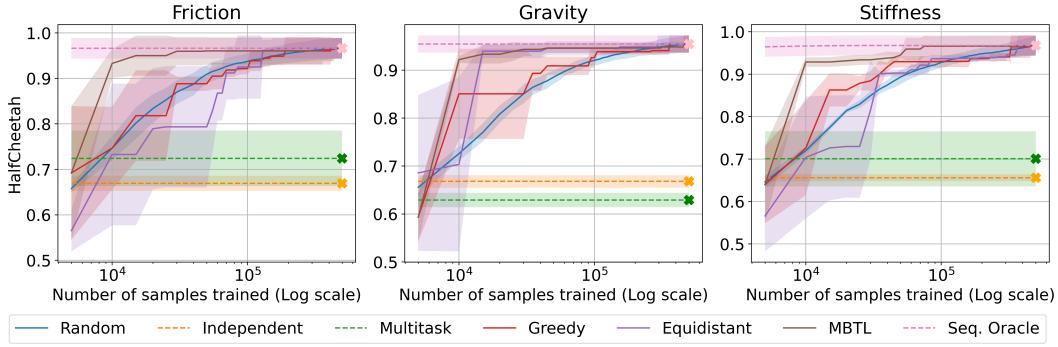


Figure 27: Comparison of normalized generalized performance of all target tasks: HalfCheetah.

## A.5 Potential impacts

Our work has the potential to reduce the computational effort needed to solve complex real-world problems, offering scalable solutions for implementing deep reinforcement learning in dynamic environments. While there are no immediate negative societal impacts identified, ongoing research will continue to assess the broader implications of deploying these technologies in urban settings.