Efficient Source Tasks Selection for Zero-shot Transfer in Contextual Reinforcement Learning

Jung-Hoon Cho
MIT
jhooncho@mit.edu

Vindula Jayawardana
MIT
vindula@mit.edu

Sirui Li
MIT
siruil@mit.edu

Cathy Wu
MIT
cathywu@mit.edu

Abstract

Deep reinforcement learning 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. This work is motivated by the empirical observation that directly applying an already trained model to a related task often works remarkably well, also called zero-shot transfer. We take this practical trick one step further to consider how to systematically select good tasks to train, maximizing overall performance across a range of tasks. Given the high cost of training, it is critical to choose a small set of training tasks. The key idea behind our approach is to explicitly model the performance loss (generalization gap) incurred by transferring a trained model. We hence introduce Model-Based Transfer Learning (MBTL) for solving contextual RL problems. In this work, we model the performance loss as a simple linear function of task context similarity. Furthermore, we leverage Bayesian optimization techniques to efficiently model and estimate the unknown training performance of the task space. We theoretically show 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 control benchmarks. Despite the conceptual simplicity, the experimental results suggest that MBTL can achieve greater performance than strong baselines, including exhaustive training on all tasks, multi-task training, and random selection of training tasks. This work lays the foundations for investigating explicit modeling of generalization, thereby enabling principled yet effective methods for contextual RL.

1 Introduction

Deep reinforcement learning (DRL) has made remarkable strides in addressing complex problems across various domains [24, 31, 1, 4, 10, 12, 23]. Despite these successes, DRL models often exhibit brittleness when exposed to small variations in task settings, significantly limiting their scalability and generalizability [28, 40]. Such variations in Markov Decision Processes (MDPs) can be conceptualized as contextual Markov Decision Processes (CMDP), where slight differences in MDP definitions lead to multiple, closely related but different environments [14, 27]. Traditional DRL approaches typically require independent training for each new task variant, a process that is both resource-intensive and impractical for real-time applications. Multi-task training is a relatively inexpensive way to train but still has limited model capacity issues. In light of this, there is a pressing need for more robust approaches that enhance generalization across different tasks. It is crucial to

17th European Workshop on Reinforcement Learning (EWRL 2024).

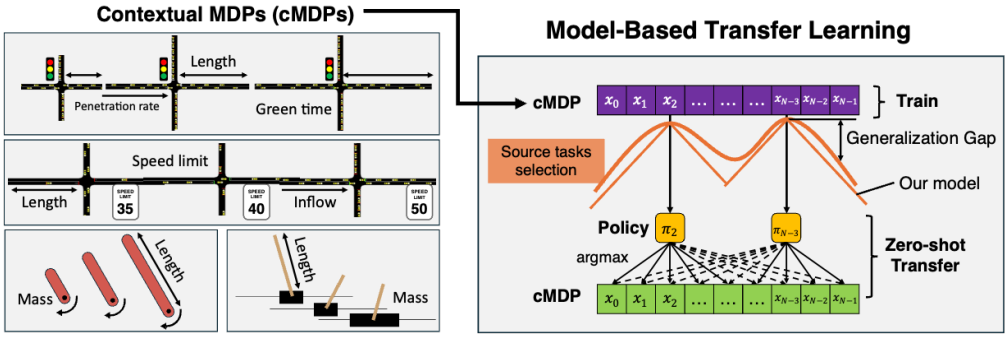


Figure 2: **Model-Based Transfer Learning (MBTL) for solving contextual MDPs**. MBTL framework strategically selects optimal tasks to train by modeling the generalization gap that arises when transferring a model to different tasks. For each target task, we deploy the most effective trained model, indicated by solid arrows. We evaluate our framework on a suite of continuous and discrete tasks, including standard control benchmarks [6] and urban traffic benchmarks [34, 19].

differentiate between *training robustness*, which seeks resilience of training across a variety of tasks, and *model robustness*, which focuses on the resistance of a model to environmental perturbations. Contextual reinforcement learning (CRL) offers a framework for modeling changes in tasks [6]. In our exploration of applying DRL to solve multiple related tasks, we identify computational challenges, particularly the high costs and complexities associated with training separate models for each task. While techniques like CARL [6] or multitask RL [36, 2] address these challenges, they still remain computationally expensive and inefficient when applied across a broad context range.

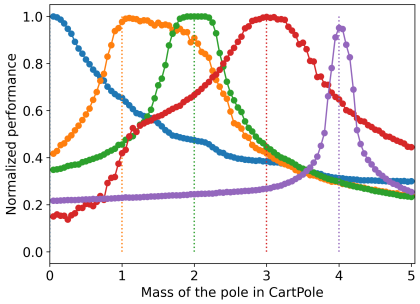


Figure 1: Generalization gap in Cartpole CMDPs. The solid lines illustrate the actual generalized performance after zero-shot transfer across contexts, with the source context indicated by a dotted line.

We build upon zero-shot transfer, which has shown potential in directly applying a policy trained in one MDP (source task) to another (target task) without adaptation. This often leads to a performance degradation known as the 'generalization gap' [16, 20]. For instance, Figure 1 exemplifies the concept of the generalization gap within a Cartpole CMDP, a range of scenarios in which the agent must balance a pole on a moving cart with a different mass of pole. The performance degrades as the target task diverges from the source task, illustrating an increasing generalization gap. Given that training is expensive yet zero-shot transfer is cheap, we are interested in optimally selecting a set of source (training) tasks, such that performance on the target range of tasks is maximized. In this article, we devise strategies to incorporate the structure of the generalization gap to better estimate the value of training. Specifically, we investigate how explicit modeling of the generalization gap structure lends itself to principled and effective algorithms for the selection of training tasks within CMDP.

Hence, this article introduces *Model-Based Transfer Learning (MBTL)*, a novel framework for studying generalization in reinforcement learning (Figure 2). We model the explicit form of the generalization gap in zero-shot transfer and use it in estimating how valuable selecting the source task is. Given a small number of samples in earlier steps, we leverage Bayesian optimization methods to estimate training performance with limited information, thus significantly reducing the necessity for exhaustive training [35, 37].

This paper contributes to the field of reinforcement learning by introducing MBTL framework for efficient source task selection, aiming to solve multiple related tasks with enhanced transferability and fewer required models. Theoretical analyses detail the sublinear cumulative regret of these methods and provide conditions for achieving tighter regret bounds. We validate our methods in simulated traffic management scenarios and standard control benchmarks. Our findings provide foundations for investigating explicit modeling of generalization, enabling efficient and practical model-based solutions for solving multiple related tasks for robust training in DRL.

2 Preliminaries and notation

We leverage the framework of CMDPs [14, 27, 6] to formulate the problem of solving multiple related tasks using reinforcement learning.

Contextual MDP (CMDP). 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 ρ is the distribution over initial states. In a CMDP, denoted by $c\mathcal{M}(x) = (S, A, P_x, R_x, \rho_x)$, these components are parameterized by a context variable $x \in X$, which can influence both dynamics and rewards [14, 27]. Consider zero-shot transfer from a source task (MDP) to solve another target task (MDP) in the CMDP. The notation x is used to represent a context for the source task, with each task x belonging to the finite set of CMDPs denoted as X. The context of the target task is similarly denoted by x' and is also an element of X. In our framework, the expected return of a MDP $c\mathcal{M}(x)$ with context x is denoted by $J(x)$. We differentiate between estimated values $\hat{J}(x)$ and observed outcomes $J(x)$, with the latter measured after training and evaluation.

Generalization gap via zero-shot transfer. Zero-shot transfer involves applying a model trained on a source task $c\mathcal{M}(x)$ to a different target task $c\mathcal{M}(x')$, with x' also belonging to the set X. We observe the *generalized performance*, denoted by $U(x'; x)$, by evaluating the target task x' based on the model trained using source task x via zero-shot generalization. As the concept of "transferability" has been discussed in literature [32, 3], we define the generalization gap as the absolute performance difference in average reward when transferring from source task x to target task x':

$$\underbrace{\Delta J(x, x')}_{\text{Generalization gap}} = \underbrace{U(x'; x)}_{\text{Generalized performance}} - \underbrace{J(x)}_{\text{Source task performance}}. \tag{1}$$

3 Problem formulation

Sequential source task selection problem. The selection of source tasks in CMDPs is key to solving the overall CMDP [3]. Specifically, we consider the *sequential source task selection (SSTS) problem*, which seeks to maximize the expected performance across a dynamically selected set of tasks. This problem is framed 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.

Definition 1 (Sequential Source Task Selection Problem)**.** *The sequential source task selection (SSTS) problem seeks to optimize the overall expected performance across a CMDP $c\mathcal{M}(\cdot)$ by selecting one task $x \in X$ at each stage. Formally, at each selection step k, we aim to choose a task x_k such that the cumulative estimated performance is maximized:*

$$\max_{x_k} \; V(x_k; x_1, ..., x_{k-1}) \quad \textit{s.t. } x_k \in X \setminus \{x_1, ..., x_{k-1}\}. \tag{2}$$

where $V(\cdot)$ is defined in the following. Intuitively, we can keep track of the highest performance achievable by any model trained thus far. Formally, after we train the k^{th} model at source task x_k, we update the *generalized performance*, denoted by $U(x'; x_{(1:k)})$, by comparing the generalized performance from the model trained using task x_k. We recursively define U based on the previous observations, as follows:

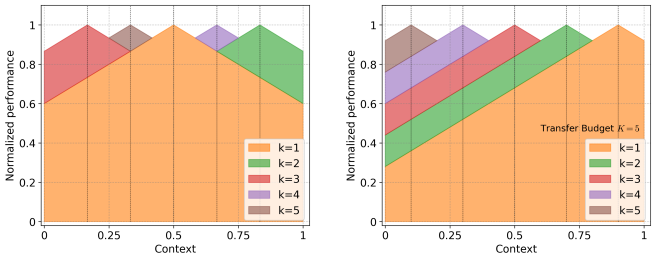
$$U(x'; x_{(1:k)}) = \max\big(U(x'; x_k), U(x'; x_{(1:k-1)})\big) \quad \forall x' \in X \quad \text{if } k > 1. \tag{3}$$

For simplicity, we denote the sequence $x_1, x_2, ..., x_k$ by $x_{(1:k)}$. We can calculate the observed expected generalized performance, denoted as $V(x_{(1:k)}) = \mathbb{E}_{x' \in X}\left[U(x'; x_{(1:k)})\right]$.

The state at each step k is defined by the best performance for each task, achieved by models trained in earlier stages, represented as $U(x'; x_{(1:k-1)})$ for each target task x'. The action at each step involves choosing a new task x_k, aimed at optimizing generalized performance based on current knowledge. The cumulative reward is quantified by the observed performance of the combined task sequence, $V(x_{(1:k)})$, providing a direct measure of the strategic value of each decision. 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 MDP x will always yield the same

Algorithm 1 Model-based Transfer Learning (MBTL)

Input: CMDPs $c\mathcal{M}(x)$, Task $x \in X$, Transfer budget K
Output: $\boldsymbol{\pi}$ and V
Initialize : $J, V = 0 \; \forall x \in X$, $\boldsymbol{\pi} = \{\}$, $k = 1$
1: **while** $k \leq K$ **do**
2: $x_k \leftarrow$ **NextTask**(J_{k-1}, V_{k-1})
3: $\pi_k \leftarrow \text{Train}(\mathcal{M}(x_k))$
4: $\boldsymbol{\pi} \leftarrow \boldsymbol{\pi} \cup \{\pi_k\}$
5: Update $J(x), V(x)$
6: $k \leftarrow k + 1$
7: **end while**
8: **return** $\boldsymbol{\pi}$ and V



(a) Pseudo-greedy strategy (b) Equidistant strategy

Figure 3: Naive methods for solving SSTS. Note that the normalized performance of the trained tasks is assumed to always be 1 (known and constant).

performance $J(x)$ and generalization gap $\Delta J(x, x'), \forall x' \in X$. The problem's horizon is defined by $|X|$, indicating that all models will be trained unless a specific termination condition is met. This termination condition is determined by reaching a predefined level of suboptimality.

Model-based transfer learning (MBTL). All methods that solve SSTS are transfer learning methods, by definition. We additionally designate a method that solves SSTS to be *model-based* if an explicit structure is imposed upon $\Delta J(\cdot, \cdot)$. To the best of the authors' knowledge, this is the first work to study such methods in the context of reinforcement learning.

4 Methods for Model-Based Transfer Learning

4.1 Modeling assumptions

We make key assumptions about the task space, performance functions, and the generalization gap.

Assumption 1 (Continuity of the task space)**.** *The task set X is continuous, meaning that for any two tasks $x, x' \in X$ and any $\epsilon > 0$, there exists a $\delta > 0$ such that $\|x - x'\| < \delta$ implies the corresponding tasks x and x' are indistinguishably close. Formally,*

$$\forall x, x' \in X, \forall \epsilon > 0, \exists \delta > 0 \text{ such that } \|x - x'\| < \delta \implies \|J(x) - J(x')\| < \epsilon.$$

Assumption 2 (Smoothness of the performance function)**.** *The performance function $J(x)$, $U(x)$, or $V(x)$ is smooth. This implies that these functions are continuously differentiable over x. Formally, $J(x), U(x), V(x) \in C^1(X)$, where $C^1(X)$ denotes the space of continuously differentiable functions on X.*

Assumption 3 (Linear generalization gap)**.** *A linear model is used to model the generalization gap function, formally $\Delta\hat{J}(x, x') \simeq \theta|x - x'|$, where θ is the slope of the linear function and x and x' are the context of the source task and target task, respectively.*

Assumption 1 and 2 are for implementing MBTL, which relies on the continuous task space and smooth performance function. Moreover, we simplify the generalization gap function as a linear function inspired by empirical observations such as shown in Figure 1. Such structure (Assumption 3) enables the development of provable algorithms for solving SSTS. Algorithm 1 provides a generic psuedocode for a MBTL algorithm. The generalization gap structure is leveraged within Line 2 (function **NextTask**).

4.2 Naive strategies

To understand what is desirable in a method for solving SSTS, it is instructive to consider some naive strategies. An **Equidistant Strategy** (ES) samples source tasks so that the tasks are equidistant along the full range of tasks we aim to solve. In this method, source tasks are selected to be equidistant within the context space based on the transfer budget K. Mathematically, the selected source tasks are chosen such that the distances between consecutive tasks are approximately equal, i.e., $x_k = \frac{2k-1}{2K}|X| \quad \text{for } k = 1, 2, \ldots, K$. Note that this strategy requires advanced knowledge

of the total number of training tasks (e.g., training budget). A **Greedy Strategy** (GS) greedily selects the source task that is predicted to have the highest generalized performance in each round, assuming known and constant training performance. Figure 3 depicts the difference between ES and GS; observe that the normalized performance of the trained tasks is assumed to be 1. However, a critical question emerges concerning scenarios where training performance is unknown or stochastic. Furthermore, ES and GS are open-loop strategies; that is, they do not incorporate feedback from actual training outcomes.

4.3 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 [26, 7]. BO integrates prior knowledge with observations to efficiently decide the next task to train, aiming to optimize the predicted value.

Gaussian Process. Within the framework of BO, we model the source training performance $J_{\text{pred}}(x)$ using Gaussian Process (GP) regression. Specifically, the function $J_{\text{pred}}(x)$ is assumed to follow a GP ($J_{\text{pred}}(x) \sim \mathcal{GP}(m(x), k(x, x'))$), where $m(x)$ represents the mean function, $\mathbb{E}[J_{\text{pred}}(x)]$, and $k(x, x')$ is the covariance function, representing the expected product of deviations of $J_{\text{pred}}(x)$ and $J_{\text{pred}}(x')$ from their respective means. Let D_{k-1} denote the data observed up to iteration $k-1$, consisting of the pairs $\{(x_i, V_{\text{obs}}(x_i))\}_{i=1,\ldots,k-1}$, where $V_{\text{obs}}(x_{(1:i)})$ are the observed outcomes. The posterior prediction of J_{pred} at a new point x, given the data D_{k-1} and previous inputs $x_{(1:k-1)}$, is normally distributed as $P(J_{\text{pred},k} \mid D_{k-1}) = \mathcal{N}(\mu_k(x), \sigma_k^2(x))$. $\mu_k(x)$ and $\sigma_k^2(x)$ are defined as $\mu_k(x) = m(x) + \mathbf{k}^\top(\mathbf{K}+\sigma^2\mathbf{I})^{-1}\mathbf{y}$ 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., $\mathbf{k} = [k(x, x_1), \ldots, k(x, x_{k-1})]$, and $\mathbf{K}$ is the covariance matrix for the observed inputs, defined as $\mathbf{K} = [k(x_i, x_j)]_{1 \le i,j \le k-1}$. This formulation 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 $x_k = \arg\max_x a(x; x_{(1:k-1)})$. Especially in our case, the acquisition function can be designed as the expected marginal improvement of generalized performance across all tasks. 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 current models (exploitation) and the measure of uncertainty associated with the task's outcome (exploration). It is defined as follows:

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

where $[\cdot]_+$ represents $\max(\cdot, 0)$ and we can use various forms of β_k.

Generalized performance. We focus on the generalized performance evaluated across all target tasks, defined as $g(x) = \mathbb{E}_{x' \in X}[J_{\text{pred}}(x) - \Delta J(x, x')] = J_{\text{pred}}(x) - \mathbb{E}_{x' \in X}[\Delta J(x, x')]$.

4.4 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 $r_k = g(x_k^*) - g(x_k)$, where $g(x_k^*)$ represents the maximum generalized performance achievable across all tasks, and $g(x_k)$ is the generalized 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. [35], our goal is to establish that this cumulative regret grows sublinearly with respect to the number of iterations. Mathematically, we aim to prove that $\lim_{K \to \infty} \frac{R_K}{K} = 0$, indicating that, on average, the performance of our strategy approaches the optimal performance as the number of iterations increases.

Regret of MBTL-GP. Having established the general framework for regret analysis, we now turn our attention to the specific regret properties of our MBTL-GP algorithm. To analyze the regret of MBTL-GP, 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 [35].

Theorem 1 (Sublinear Regret)**.** *Given $\delta \in (0,1)$, and with the scaling factor β_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 $Pr\left[R_K \leq \sqrt{KC_1\beta_K\gamma_K}\right] \geq 1-\delta$, 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 model'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 simple linear function in Assumption 3. 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. Consequently, at each step, we effectively reduce the search space.

Given the generalization gap observed in Figure 1, we model the degradation from the source task using a simple linear function in Assumption 3. 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.

Definition 2 (Reduced search space)**.** *We define the reduced search space X_k as a subset of X, with each element $x' \in X_k$, such that $U(x'; x_{(1:k-1)}) \leq \hat{J}(x_k) + \Delta J(x', x_k)$.*

We leverage the reduced uncertainty in well-sampled regions to tighten the regret bound while slightly lowering the probability δ in Theorem 1. For the regret analysis, we propose the following theorem based on the generalization of Lemma 5.2 and 5.4 in [35] to the eliminated search space.

Theorem 2. *For a given $\delta' \in (0,1)$ and scaling factor $\beta_k = 2\log(|X|\pi^2k^2/6\delta)$, 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.1.2.

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.1.3 and A.1.4, respectively. Based on our experiments presented in Section 5, the rate of elimination of the largest segment for MBTL-GP is shown in Figure 4.

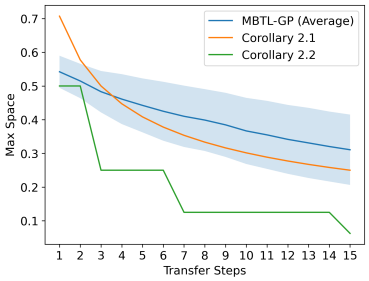


Figure 4: Empirical results of the restriction of search space by MBTL-GP compared to two examples from Corollaries 2.1 and 2.2.

Table 1: Comparative performance of different methods on traffic CMDP tasks

Benchmark		Baselines			MBTL (Ours)			Oracle
Task	**Context Variation**	**Random**	**Exhaustive**	**Multitask**	**MBTL-GS**	**MBTL-ES**	**MBTL-GP**	**Oracle Transfer**
Number of Trained Models		k	N	1	k	K	k	N
Traffic Signal	Road Length	0.9249	**0.9409**	0.8242	0.9278	0.9213	0.9371	0.9409
Traffic Signal	Inflow	0.8457	0.8646	0.8319	0.8496	**0.8700**	0.8673	0.8768
Traffic Signal	Speed Limit	0.8821	0.8857	0.6083	**0.8862**	0.8858	0.8854	0.8876
Eco-Driving	Penetration Rate	0.5959	0.5260	0.1945	0.5827	0.5934	**0.6323**	0.6660
Eco-Driving	Inflow	0.4774	0.4061	0.2229	0.4673	0.4705	**0.5108**	0.5528
Eco-Driving	Green Phase	0.4406	0.3850	0.4228	0.4431	0.4557	**0.4700**	0.5027
AA-Ring-Acc	Hold Duration	0.8924	0.8362	0.9209	0.8776	0.9057	**0.9242**	0.9552
AA-Ring-Vel	Hold Duration	0.9785	0.9589	0.9720	0.9807	0.9772	**0.9816**	0.9822
AA-Ramp-Acc	Hold Duration	0.6050	0.4276	0.5158	0.6143	0.5956	**0.6318**	0.7111
AA-Ramp-Vel	Hold Duration	0.6690	0.5473	0.5034	0.5907	0.6787	**0.7182**	0.7686
Average		0.7312	0.6778	0.6017	0.7220	0.7354	**0.7559**	0.7844

†Higher the better. The bold indicates the best for each benchmark, excluding the Oracle transfer. Detailed results with variance for each method are provided in Appendix A.2.2.
‡AA: Advisory autonomy tasks, Ring: Single lane ring, Ramp: Highway ramp, Acc: Acceleration guidance, Vel: Speed guidance.

5 Experiments and analysis

5.1 Setup

Our experiments span a variety of CMDP settings, focusing on transportation tasks as well as classic control benchmarks. We introduce three intelligent transportation tasks that utilize DRL but suffer from brittleness issues, as well as the classic control tasks.

Baselines. Our proposed methods are compared against a range of baselines including (1) **Random selection**, where each training task is chosen randomly; (2) **Exhaustive training**, which involves independent training separate models on all tasks exhaustively but is computationally expensive; (3) **Multi-task RL**, where a single context-conditioned model is trained for all tasks; (4) **Oracle transfer**, which assumes full knowledge of generalized performance for all policies and selects the best source task for each target task.

Proposed methods. Our proposed methods are: (1) **MBTL-ES**, selecting source tasks for the context to be equidistance based on the transfer budget; (2) **MBTL-GS**, selecting the next source task greedily to maximize the marginal improvement of expected generalized performance based on linear generalization gap (Assumption 3); (3) **MBTL-GP**, selecting source task based on the BO using GP and UCB with 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 actions [24] and Proximal Policy Optimization (PPO) for continuous actions [30]. We evaluate our methods by the average performance across all target tasks after training 15 source tasks and the number of trained models needed to achieve a certain level of suboptimality. We employ min-max normalization of the rewards for each task.

5.2 Traffic tasks

We validate our methods on diverse simulated traffic scenarios, focusing on to what extent Our proposed methods can optimize the global objective only with a small number of trained models.First, while most traffic lights operate on fixed schedules, we can design learning-based adaptive (1) **Traffic signal control** to optimize the traffic [8, 21]. However, considering that every intersection looks different, challenges persist in generalizing across various intersection configurations [18]. Given the significant portion of greenhouse gas emissions in the United States coming from the transportation sector [11], the second traffic task is (2) **Eco-driving Lagrangian control**, which is critical for climate change mitigation. DRL-based eco-driving strategies have been developed [13, 38, 17] but still have some issues of difficulties in generalization. The last traffic task is (3) **Advisory autonomy**, which is a real-time speed advisory system that enables human drivers to emulate the system-level performance of autonomous vehicles in mixed autonomy systems [34, 15]. Instead of direct and instantaneous control, human drivers receive periodic guidance, which varies based on the traffic states. In Appendix A.2, we provide comprehensive details about our experiments.

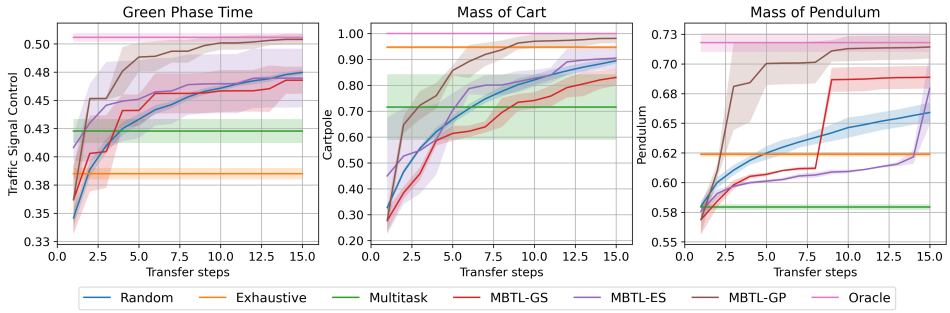


Figure 5: Comparison of normalized generalized performance of all target tasks.

Table 2: Comparative performance of different methods on standard control CMDP tasks

Benchmark		**Baselines**			**MBTL (Ours)**			**Oracle**
Task	**Context Variation**	**Random**	**Exhaustive**	**Multitask**	**MBTL-GS**	**MBTL-ES**	**MBTL-GP**	**Oracle Transfer**
Number of Trained Models		k	N	1	k	K	k	N
Pendulum	Length	0.7270	0.7383	0.6830	0.7327	0.7092	**0.7697**	0.7969
Pendulum	Mass	0.6329	0.6237	0.5793	0.6408	0.6092	**0.6827**	0.7132
Pendulum	Timestep	0.7989	0.8135	0.7247	**0.8177**	0.7541	0.8141	0.8801
Cartpole	Mass of Cart	0.7221	**0.9466**	0.7153	0.6501	0.7516	0.8212	0.9838
Cartpole	Length of Pole	0.8121	0.9110	0.5441	0.8217	0.8428	**0.9124**	0.9875
Cartpole	Mass of Pole	0.8858	**0.9560**	0.6073	0.8744	0.7909	0.9351	1.0000
BipedalWalker	Gravity	0.9330	0.9281	0.7898	0.9359	**0.9494**	0.9393	0.9674
BipedalWalker	Friction	0.9650	0.9317	0.9051	0.9645	0.9664	**0.9713**	0.9778
BipedalWalker	Scale	0.8605	0.8694	0.7452	0.8792	0.8496	**0.8886**	0.9107
HalfCheetah	Gravity	0.8542	0.6679	0.6292	0.8663	0.8634	**0.9073**	0.9544
HalfCheetah	Friction	0.8567	0.6693	0.7242	0.8591	0.8703	**0.9274**	0.9663
HalfCheetah	Stiffness	0.8533	0.6561	0.7007	0.8785	0.7817	**0.9146**	0.9674
Average		0.8251	0.8093	0.6957	0.8267	0.8116	**0.8736**	0.9255

†Higher the better. The bold indicates the best for each benchmark, excluding the Oracle transfer. Detailed results with variance for each method are provided in Appendix A.2.2.

Discussion. Table 1 compares the min-max normalized generalized performance across the full target task range. While exhaustive training is computationally expensive and exhibits suboptimal performance, the Oracle transfer, which assumes perfect knowledge of transferability, demonstrates the potential benefits of zero-shot transfer for solving the complete range of multiple tasks. Our proposed MBTL methods leverage this transferability by intelligently selecting the next source task to learn from. Notably, MBTL-GP shows superior performance, closely aligning with the Oracle transfer, which suggests the Gaussian Process effectively models the underlying task dynamics, facilitating better generalization from learned experiences. For example, MBTL-GP significantly improves generalized performance with a few source tasks for traffic signal control problems varying green phase durations (Figure 5). More results are provided in Appendix A.2.

5.3 Standard control benchmarks

Building on the evaluation of a suite of traffic-related tasks, we extend our investigation to classic control tasks to assess the applicability of our methods across a broader spectrum of environments. We utilize context-extended versions of standard RL environments from CARL benchmark library [6] to rigorously test the robustness and adaptability of our methods under varied contexts. For Cartpole tasks, we explored contextual Markov decision processes (CMDPs) with varying cart masses, pole lengths, and pole masses. We vary the timestep duration, pendulum length, and pendulum mass in Pendulum. The BipedalWalker was tested under varying friction, gravity, and scale. In HalfCheetah tasks, 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, enabling a comprehensive analysis of transfer learning under drastically different conditions. We provide more details in Appendix A.2.

Discussion. Table 2 compares the performance across the full target task range. Our experiments demonstrate that MBTL significantly improves performance and sample efficiency in dynamic urban transportation systems compared to the baselines. In control tasks such as Cartpole, Pendulum, BipedalWalker, and HalfCheetah, where physical parameters like mass directly influence the dynam-

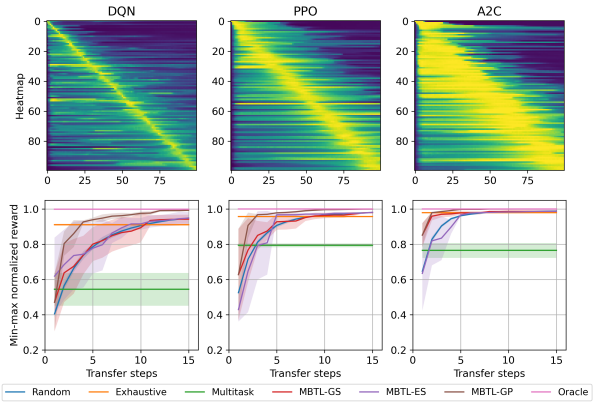


Figure 6: Sensitivity analysis on different DRL algorithms (Cartpole, length of pole).

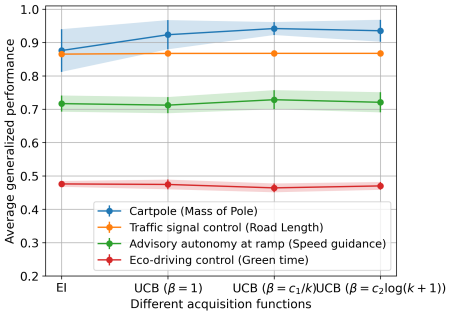


Figure 7: Sensitivity analysis on the different acquisition functions.

ics, MBTL methods, particularly MBTL-GP, excel by effectively transferring knowledge between similar task variations. For example, in Figure 5, MBTL-GP outperforms other baselines, including multitask training. The results underscore the potential of MBTL in solving multiple related tasks.

5.3.1 Sensitivity analysis

DRL algorithms. Figure 6 shows an ablation study comparing different DRL algorithms—DQN, PPO, and Advantage Actor-Critic (A2C) [25]. The heatmap results show differences in transferability, and the results underscore the robustness of MBTL to different algorithms.

Acquisition functions. Figure 7 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 exploitation. For Cartpole and advisory autonomy tasks, UCB with decreasing β_k as k increases shows the best performance. However, for eco-driving control, the β_k formulation in Section 4.4 exhibits the highest performance, though the difference is not statistically significant.

6 Related work

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 [14, 27, 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 [14, 27].

The contexts of CMDPs are not always visible during training [20]. When they are visible, they can be directly used as side information by conditioning the policy on them [33]. A common approach is multi-task learning, which aims to leverage a 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 [36]. However, when the context features are not visible, the resulting CMDP becomes a partially observable CMDP [20, 9] and can be challenging for multi-task learning. In this work, we use multi-task learning as a baseline in evaluating our methods.

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 [20]. Rezaei-Shoshtari et al. [29] 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. [32] uses meta-data to learn inter-task transferability to learn the expected benefit of transfer given a source-target task pair.

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. [39] 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.

7 Conclusion

Our study refines the robustness of DRL by developing MBTL using zero-shot transfer. MBTL strategically selects a set of source tasks to maximize overall performance and minimize training costs. It models the generalization gap and uses BO to predict task performance efficiently. We theoretically show that the method exhibits regret that is sublinear in the number of training tasks. We validate MBTL with urban traffic and control benchmarks. **Limitation** is that MBTL relies highly on assumptions that may not hold in highly dynamic or complex environments, potentially limiting its applicability and leading to suboptimal policy convergence. Future work will focus on improving the modeling of the generalization gap and exploring scenarios with multiple context variations where simple strategies like MBTL-ES or MBTL-GS may not be applicable.

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] 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.

[9] 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.

[10] 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.

[11] US EPA. Sources of Greenhouse Gas Emissions, 2023. URL `https://www.epa.gov/ghgemissions/sources-greenhouse-gas-emissions`.

[12] Alhussein Fawzi, Matej Balog, Aja Huang, Thomas Hubert, Bernardino Romera-Paredes, Mohammadamin Barekatain, Alexander Novikov, Francisco J R Ruiz, Julian Schrittwieser, Grzegorz Swirszcz, et al. Discovering faster matrix multiplication algorithms with reinforcement learning. *Nature*, 610(7930):47–53, 2022.

[13] 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.

[14] Assaf Hallak, Dotan Di Castro, and Shie Mannor. Contextual markov decision processes. *arXiv preprint arXiv:1502.02259*, 2015.

[15] Aamir Hasan, Neeloy Chakraborty, Haonan Chen, Jung-Hoon Cho, Cathy Wu, and Katherine Driggs-Campbell. Perp: Personalized residual policies for congestion mitigation through co-operative advisory systems. In *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, pages 5078–5085. IEEE, 2023.

[16] 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.

[17] Vindula Jayawardana and Cathy Wu. Learning eco-driving strategies at signalized intersections. In *2022 European Control Conference (ECC)*, pages 383–390. IEEE, 2022.

[18] 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.

[19] Vindula Jayawardana, Sirui Li, Cathy Wu, Yashar Farid, and Kentaro Oguchi. Generalizing Cooperative Eco-driving via Multi-residual Task Learning, March 2024.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] Aravind Rajeswaran, Vikash Kumar, Abhishek Gupta, Giulia Vezzani, John Schulman, Emanuel Todorov, and Sergey Levine. Learning Complex Dexterous Manipulation with Deep Reinforcement Learning and Demonstrations, June 2018.

[29] 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.

[30] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal Policy Optimization Algorithms, August 2017.

[31] 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.

[32] 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.

[33] 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.

[34] 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.

[35] 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.

[36] 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.

[37] Hung Tran-The, Sunil Gupta, Santu Rana, Huong Ha, and Svetha Venkatesh. Sub-linear Regret Bounds for Bayesian Optimisation in Unknown Search Spaces. In *Advances in Neural Information Processing Systems*, volume 33, pages 16271–16281. Curran Associates, Inc., 2020.

[38] 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.

[39] 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.

[40] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Avnish Narayan, Hayden Shively, Adithya Bellathur, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-World: A Benchmark and Evaluation for Multi-Task and Meta Reinforcement Learning, June 2021.

A Appendix

Contents

A.1 Theoretical analysis

A.1.1 Notation

Table 3 describes the notation used in this paper.

Table 3: Notation used in the problem formulation

Symbol	Description
x	Source task ($x \in X$)
x'	Target task ($x' \in X$)
x_k	Selected source task at transfer step k ($k = 1, ..., K$)
$\mathbf{x}$	Vector of source tasks ($\mathbf{x} = [x_1, ..., x_k]$ where $x_i \in X$ for $1 \leq i \leq k$)
$c\mathcal{M}(x)$	Contextual MDP parameterized by x
$J(x)$	Performance of task $c\mathcal{M}(x)$
$\Delta\hat{J}(x, x')$	Generalization gap (source: x, target: x')
$U(x'; x)$	Transferred performance (source: x (or $\mathbf{x}$), target: x')
$V(x)$	Expected generalized performance of source model x evaluated on all $x' \in X$

Figure 8 helps understand the discrepancy between the observed generalized performance and the predicted one. Figure 9 illustrates how to calculate the marginal improvement of expected generalized performance ($\hat{V}(x; x_1, ..., x_{k-1}) - V(x_1, ..., x_{k-1})$).

A.1.2 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 $1 - \delta'$.*

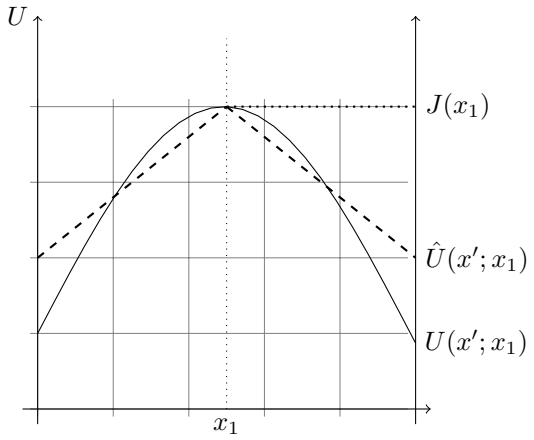


Figure 8: Illustration of the discrepancy between observed (U) and predicted ($\hat{U}$) generalized performance after training on source task x_1 and attempting zero-shot transfer to x'.

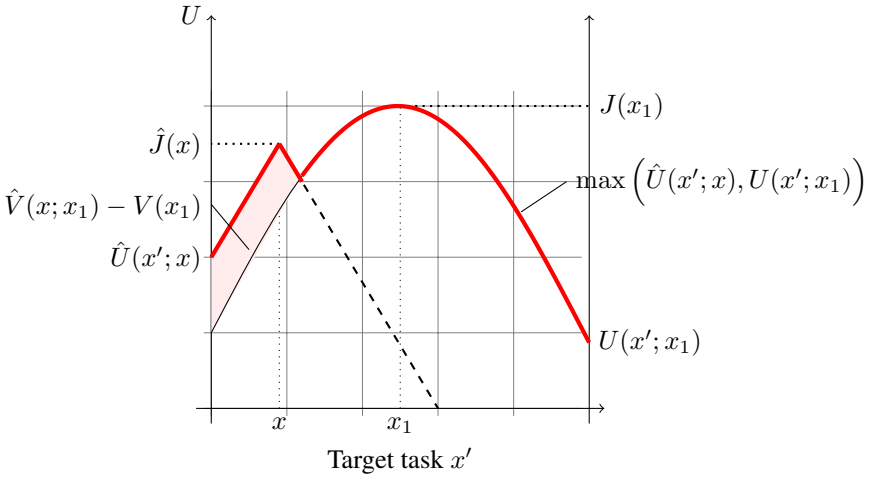


Figure 9: Step for choosing x_2 that maximizes the estimated marginal improvement ($\hat{V}(x; x_1) - V(x_1)$). $\hat{V}(x; x_1)$ corresponds to the red area under the red line and $V(x_1)$ as the area under $U(x'; x_1)$.

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 [35].

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 [35], Lemma 4, and Cauchy–Schwarz inequality, we can bound the cumulative regret as:

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

□

A.1.3 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:

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

The cumulative regret can be bounded as below:

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

□

A.1.4 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:

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

The cumulative regret can be bounded as below:

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

□

A.2 Experiment details

A.2.1 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 $\{5, 10, 15\}$. We explored several kernel configurations, including the combination of a constant kernel (C) and a radial basis function kernel (RBF) with a length scale ranging from 0.01 to 100, as well as more complex kernels incorporating a white noise kernel and an exponential sine squared kernel with varying length scales and periodicities. Each GP model was instantiated with the specified kernel, and the noise level was set to the square of the noise standard deviation. This systematic exploration aimed to identify the optimal set of hyperparameters that could enhance the model's predictive accuracy.

A.2.2 Results of table with standard deviation

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

Benchmark		**Baselines**			**Model-Based Transfer Learning**			**Oracle**
Task	**Context Variation**	**Random**	**Exhaustive**	**Multitask**	**MBTL-PS**	**MBTL-ES**	**MBTL-GP**	**Oracle Transfer**
Number of Trained Models		k	N	1	k	K	k	N
Traffic Signal	Road Length	0.9249 (0.0007)	**0.9409** (0.0002)	0.8242 (0.0538)	0.9278 (0.0071)	0.9213 (0.0001)	0.9371 (0.0023)	0.9409 (0.0006)
Traffic Signal	Inflow	0.8457 (0.0028)	0.8646 (0.0009)	0.8319 (0.004)	0.8496 (0.0039)	**0.8700** (0.0051)	0.8673 (0.0016)	0.8768 (0.0012)
Traffic Signal	Speed Limit	0.8821 (0.0003)	0.8857 (0.0005)	0.6083 (0.0403)	**0.8862** (0.001)	0.8858 (0.0004)	0.8854 (0.0008)	0.8876 (0.0002)
Eco-Driving	Penetration Rate	0.5959 (0.0058)	0.5260 (0.0087)	0.1945 (0.0057)	0.5827 (0.0151)	0.5934 (0.0052)	**0.6323** (0.0235)	0.6660 (0.0049)
Eco-Driving	Inflow	0.4774 (0.0097)	0.4061 (0.0094)	0.2229 (0.001)	0.4673 (0.0124)	0.4705 (0.0051)	**0.5108** (0.0175)	0.5528 (0.0096)
Eco-Driving	Green Phase	0.4406 (0.0045)	0.3850 (0.0063)	0.4228 (0.0184)	0.4431 (0.0109)	0.4557 (0.0334)	**0.4700** (0.0144)	0.5027 (0.0058)
AA-Ring-Acc	Hold Duration	0.8924 (0.0097)	0.8362 (0.0048)	0.9209 (0.0235)	0.8776 (0.0103)	0.9057 (0.0338)	**0.9242** (0.021)	0.9552 (0.0132)
AA-Ring-Vel	Hold Duration	0.9785 (0.0012)	0.9589 (0.0096)	0.9720 (0.0035)	0.9807 (0.0007)	0.9772 (0.0072)	**0.9816** (0.0004)	0.9822 (0)
AA-Ramp-Acc	Hold Duration	0.6050 (0.0192)	0.4276 (0.0066)	0.5158 (0.045)	0.6143 (0.0793)	0.5956 (0.1052)	**0.6318** (0.0272)	0.7111 (0.0479)
AA-Ramp-Vel	Hold Duration	0.6690 (0.0476)	0.5473 (0.0222)	0.5034 (0.014)	0.5907 (0.0407)	0.6787 (0.0544)	**0.7182** (0.0461)	0.7686 (0.0581)
Pendulum	Length	0.7270 (0.0008)	0.7383 (0.0034)	0.6830 (0.0008)	0.7327 (0.005)	0.7092 (0.0027)	**0.7697** (0.013)	0.7969 (0.0123)
Pendulum	Mass	0.6329 (0.0077)	0.6237 (0.0023)	0.5793 (0.0041)	0.6408 (0.0073)	0.6092 (0.0013)	**0.6827** (0.0239)	0.7132 (0.0129)
Pendulum	Timestep	0.7989 (0.0073)	0.8135 (0.0103)	0.7247 (0.0488)	**0.8177** (0.0024)	0.7541 (0.0231)	0.8141 (0.0032)	0.8801 (0.0198)
Cartpole	Mass of Cart	0.7221 (0.0188)	**0.9466** (0.0065)	0.7153 (0.2195)	0.6501 (0.0268)	0.7516 (0.0877)	0.8212 (0.0164)	0.9838 (0.0131)
Cartpole	Length of Pole	0.8121 (0.0156)	0.9110 (0.0065)	0.5441 (0.1614)	0.8217 (0.098)	0.8428 (0.1167)	**0.9124** (0.0266)	0.9875 (0.0047)
Cartpole	Mass of Pole	0.8858 (0.0052)	**0.9560** (0.0128)	0.6073 (0.0948)	0.8744 (0.0804)	0.7909 (0.0548)	0.9351 (0.0403)	1.0000 (0.0000)
BipedalWalker	Gravity	0.9330 (0.0025)	0.9281 (0.0034)	0.7898 (0.0928)	0.9359 (0.0172)	**0.9494** (0.0119)	0.9393 (0.0123)	0.9674 (0.0014)
BipedalWalker	Friction	0.9650 (0.0021)	0.9317 (0.0074)	0.9051 (0.0734)	0.9645 (0.0098)	0.9664 (0.004)	**0.9713** (0.0008)	0.9778 (0.0013)
BipedalWalker	Scale	0.8605 (0.0081)	0.8694 (0.0087)	0.7452 (0.0938)	0.8792 (0.0186)	0.8496 (0.0141)	**0.8886** (0.0183)	0.9107 (0.004)
HalfCheetah	Gravity	0.8542 (0.015)	0.6679 (0.0162)	0.6292 (0.0258)	0.8663 (0.0563)	0.8634 (0.0493)	**0.9073** (0.0133)	0.9544 (0.0221)
HalfCheetah	Friction	0.8567 (0.036)	0.6693 (0.0203)	0.7242 (0.1056)	0.8591 (0.0397)	0.8703 (0.033)	**0.9274** (0.0392)	0.9663 (0.0276)
HalfCheetah	Stiffness	0.8533 (0.0163)	0.6561 (0.0101)	0.7007 (0.1126)	0.8785 (0.0138)	0.7817 (0.0419)	**0.9146** (0.0189)	0.9674 (0.0286)
Average		0.8068 (0.1551)	0.7775 (0.197)	0.6337 (0.2395)	0.8037 (0.1629)	0.7996 (0.1548)	**0.8435** (0.1501)	0.8833 (0.1467)

* *Note*: The bold values indicate the maximum value for each task excluding the oracle. Standard deviation across multiple runs in the parenthesis.

A.2.3 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 [8, 21], though challenges persist in generalizing across various intersection configurations [18].

Figure 10 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) [22] v.1.16.0 and PPO for RL algorithm [30]. All experiments are done on a distributed computing cluster equipped with 48 Intel Xeon Platinum 8260 CPUs.

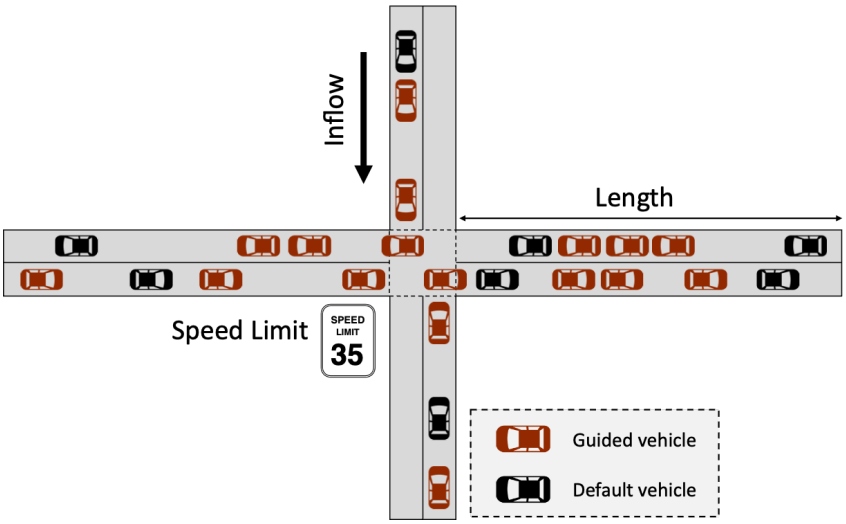
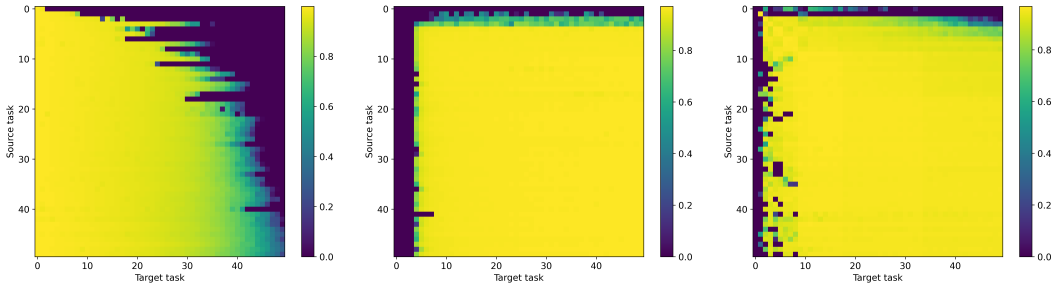


Figure 10: Illustration of the traffic networks in traffic signal control task.

Transferability heatmap Figure 11 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.



(a) Inflow variation (b) Speed limit variation (c) Road length variation

Figure 11: Examples of transferability heatmap for traffic signal control.

Results Figure 12 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-GP 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-GP, indicating robustness in adapting to different road lengths.

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

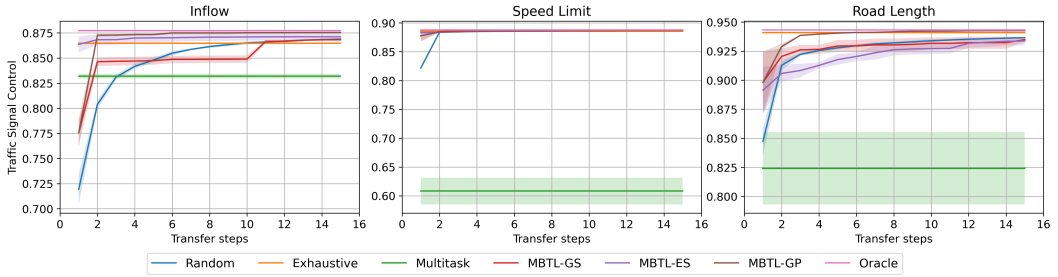


Figure 12: Comparison of normalized generalized performance of all target tasks: Traffic signal control.

A.2.4 Details about eco-driving Lagrangian control

Given the significant portion of greenhouse gas emissions in the United States coming from the transportation sector [11], eco-driving behaviors are critical for climate change mitigation. Deep reinforcement learning-based eco-driving strategies have been developed [13, 38, 17] 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 13 illustrates the traffic road network used in eco-driving Lagrangian control task. The road network is depicted with 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) [22] v.1.16.0 and PPO for RL algorithm [30].

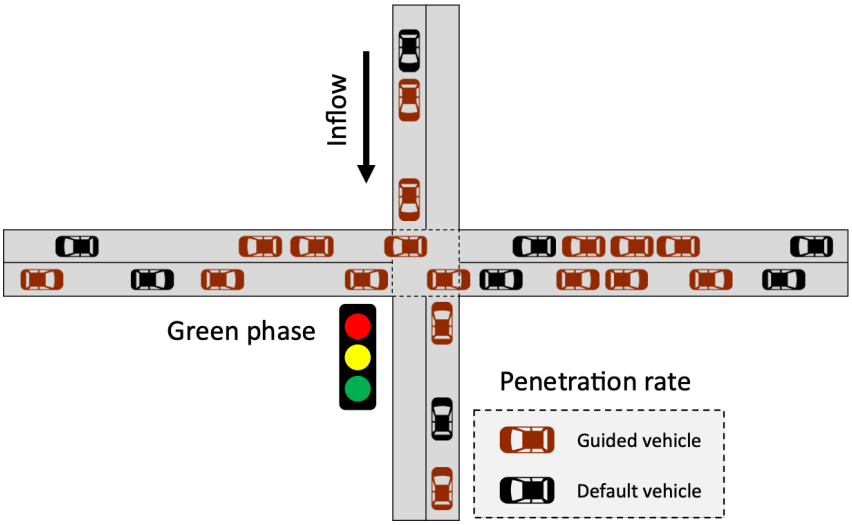
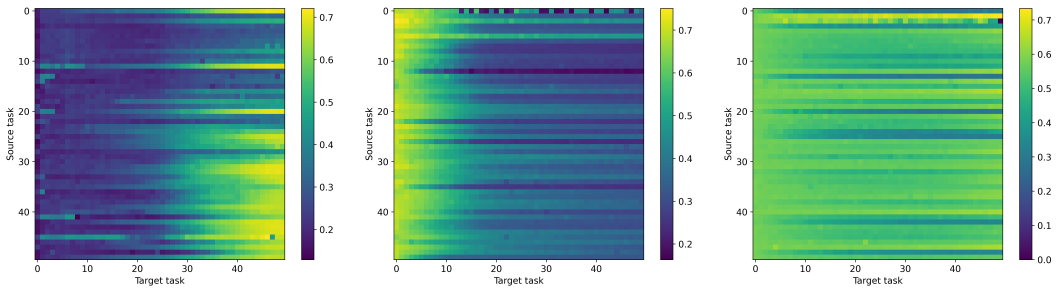


Figure 13: Illustration of the traffic networks in eco-driving control task.

Transferability heatmap Figure 14 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 15 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-GP consistently demonstrates superior performance across all variations, indicating robust adaptability to changing task parameters.



(a) Green phase variation (b) Inflow variation (c) Penetration rate variation

Figure 14: Examples of transferability heatmap for eco-driving Lagrangian control.

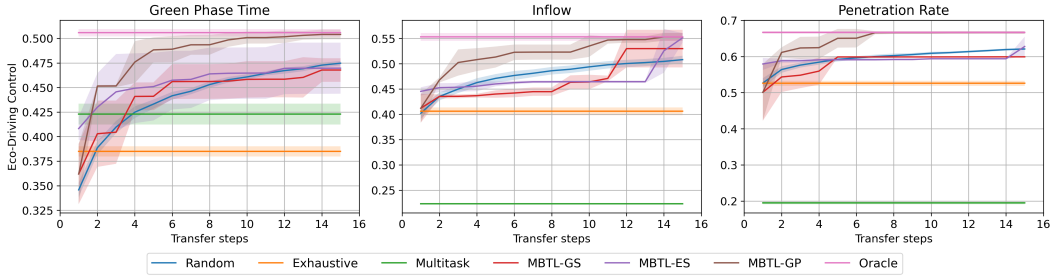


Figure 15: Comparison of normalized generalized performance of all target tasks: Eco-driving Lagrangian control.

A.2.5 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 [34, 15]. 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 16 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.

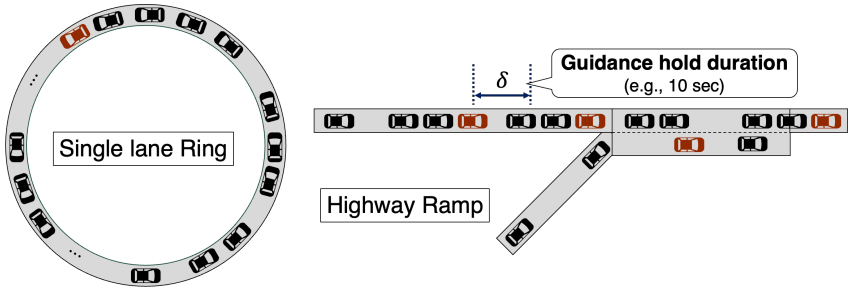
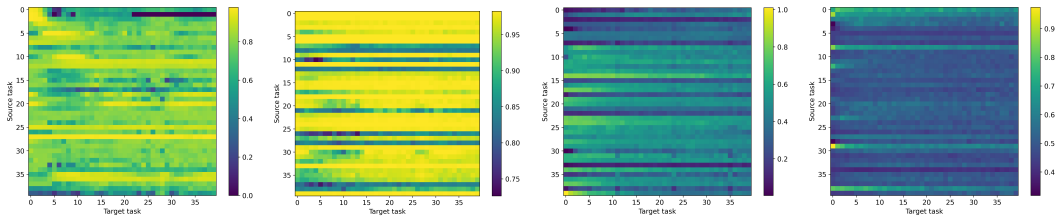


Figure 16: Illustration of the traffic networks in advisory autonomy task.

Transferability heatmap Figure 17 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.



(a) Ring with acceleration guide (b) Ring with speed guide (c) Ramp with acceleration guide (d) Ramp with speed guide

Figure 17: Examples of transferability heatmap for advisory autonomy.

Results Figure 18 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-GP 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-GP closely matching in some instances.

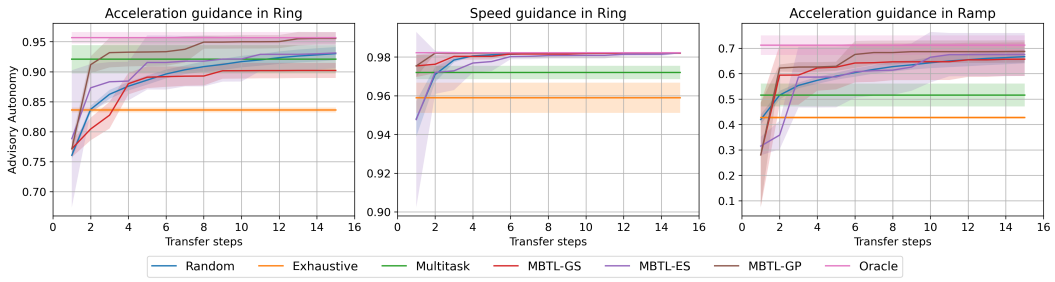


Figure 18: Comparison of normalized generalized performance of all target tasks: Advisory autonomy.

A.2.6 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.2.7 Results of control tasks

Figure 19 compares our proposed methods with baselines and oracle method in different variations of context in classic control tasks.

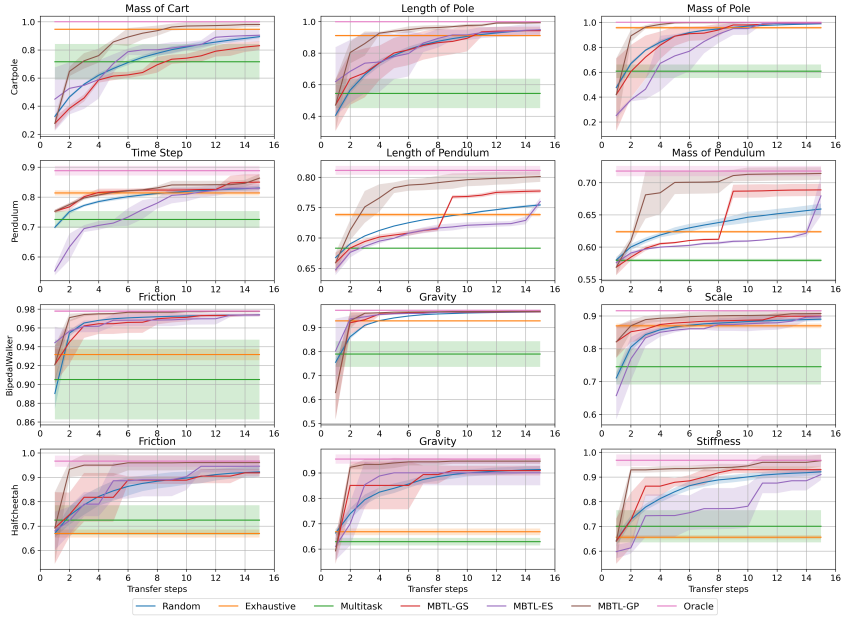
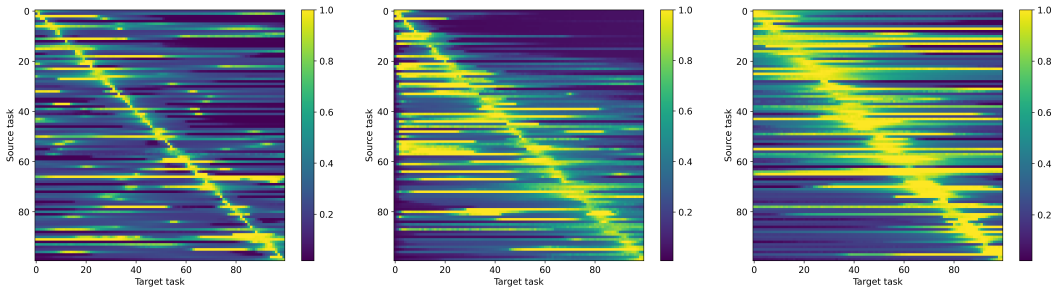


Figure 19: Comparison on control benchmarks: Cartpole, Pendulum, BipedalWalker, and Halfcheetah.

A.2.8 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.



(a) Mass of cart variation (b) Length of the pole variation (c) Mass of pole variation

Figure 20: Examples of transferability heatmap for Cartpole.

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-GP 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-GP shows close to oracle performance.

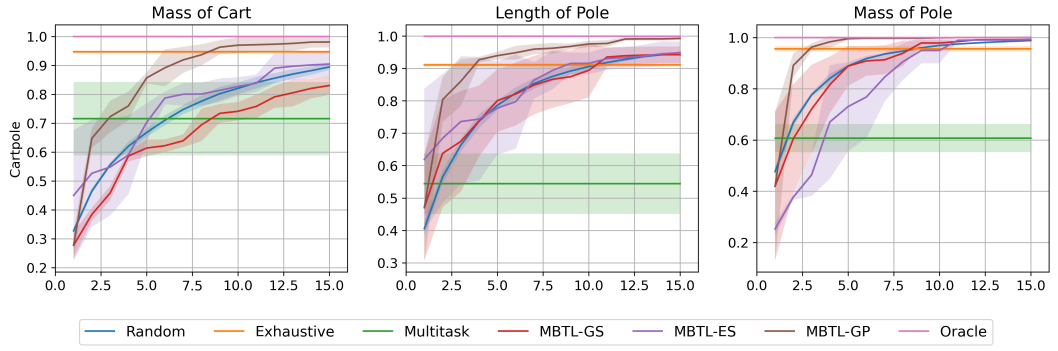
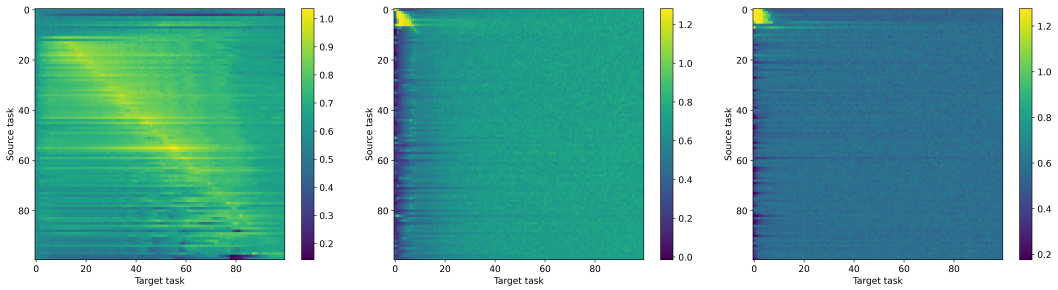


Figure 21: Comparison of normalized generalized performance of all target tasks: Cartpole.

A.2.9 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.



(a) Timestep variation

(b) Length of the pendulum variation

(c) Mass of pendulum variation

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-GP strategies demonstrate the highest scores, suggesting robust adaptability to changes in pendulum dynamics. MBTL-GP shows performance close to that of the Oracle across all variations, indicating its effectiveness in handling dynamic changes in system parameters.

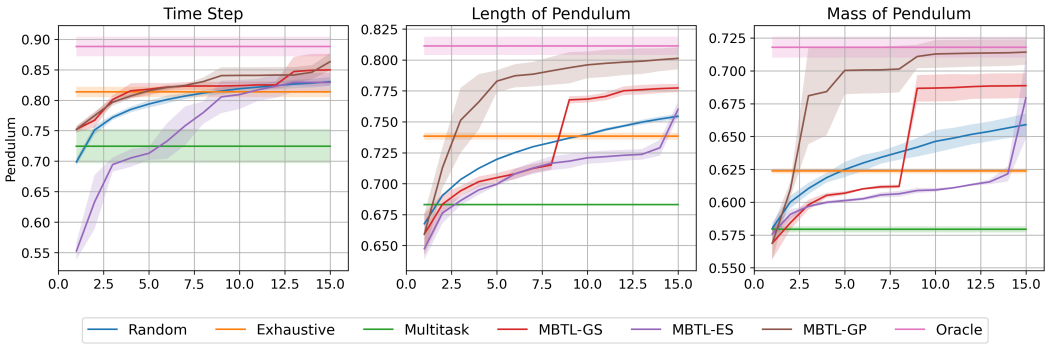
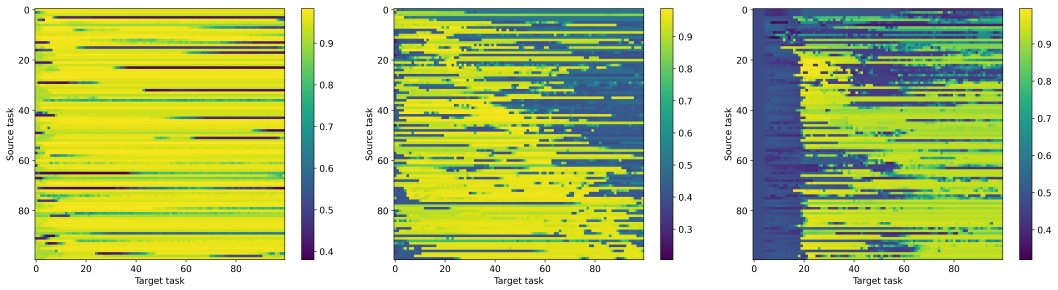


Figure 23: Comparison of normalized generalized performance of all target tasks: Pendulum.

A.2.10 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.



(a) Friction variation (b) Gravity variation (c) Scale variation

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 tabualr results in Table 2, MBTL-GP 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.

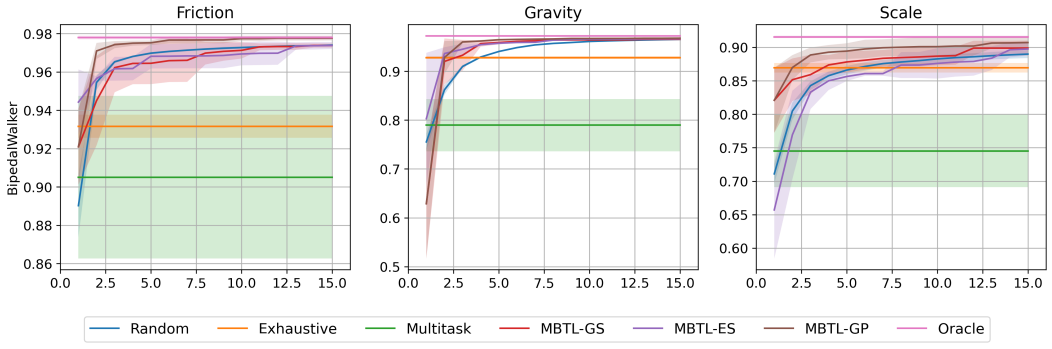
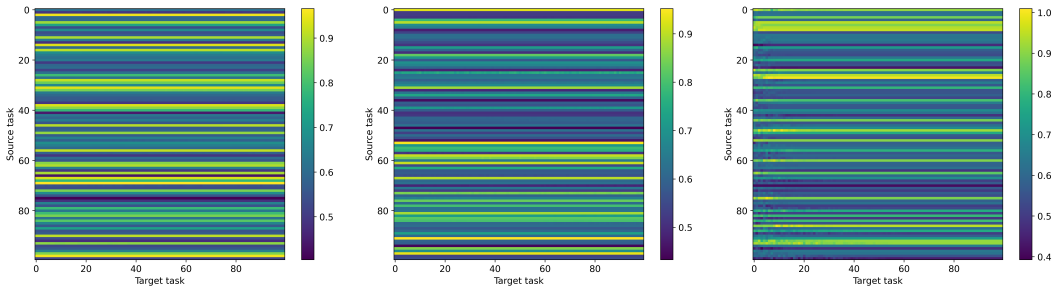


Figure 25: Comparison of normalized generalized performance of all target tasks: BipedalWalker.

A.2.11 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.



(a) Friction variation (b) Gravity variation (c) Stiffness variation

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-GP 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.

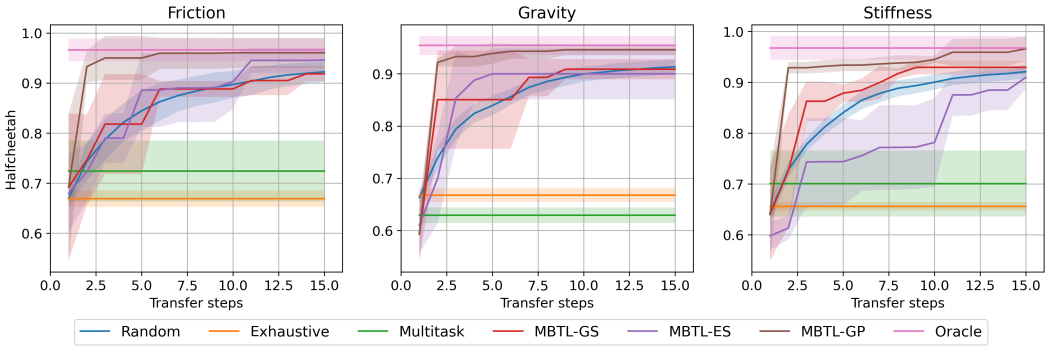


Figure 27: Comparison of normalized generalized performance of all target tasks: HalfCheetah.

A.3 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.