Model-Based Transfer Learning for 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 (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 43x 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/.

1 Introduction

Deep reinforcement learning (DRL) has made remarkable strides in addressing complex problems across various domains [29, 37, 1, 4, 10, 12, 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 [14, 32, 5].

There are two predominant solution modalities for CMDPs [23]: independent training and multi-task training. Independent training constructs a separate model for each task variant (say, $N \gg 1$), which

38th Conference on Neural Information Processing Systems (NeurIPS 2024).
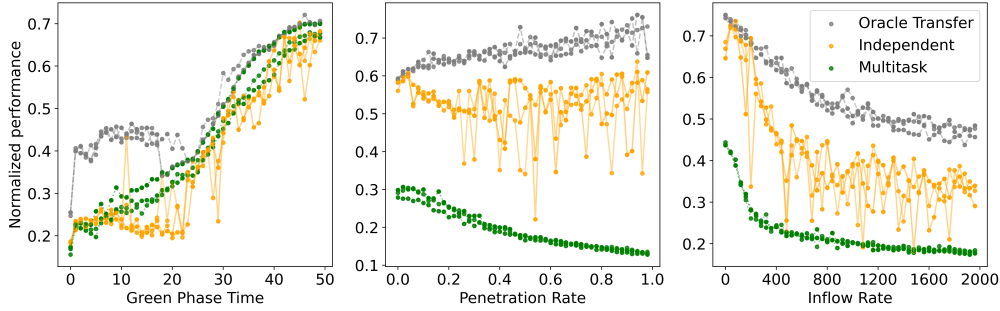
Figure 1: Normalized performance comparison across different problem variations in Eco-Driving benchmark. Traditional DRL approaches (e.g., Independent training or multi-task training) exhibit greater training instability, whereas Oracle Transfer, zero-shot transfer with full information, shows the potential for performance improvement by multi-policy training.

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 [22, 46, 2, 42]. 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*.

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. Figure 1 shows that multi-policy training with zero-shot transfer has the potential to improve the performance even over the independent training. In this article, we strategically select source tasks by explicitly modeling the generalization performance to estimate the value of training a new source task.

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.

The main contributions of this work are:

- We introduce *Model-Based Transfer Learning (MBTL),* a novel framework for solving CMDP sample efficiently (Figure 2). To the best of our knowledge, this is the first work to explicitly model generalization performance for contextual RL (CRL). As such, our work opens the door for further investigation into reliable model-based methodologies for CRL.
- 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 43x** 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 ρ is the distribution over initial states [45]. 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

2
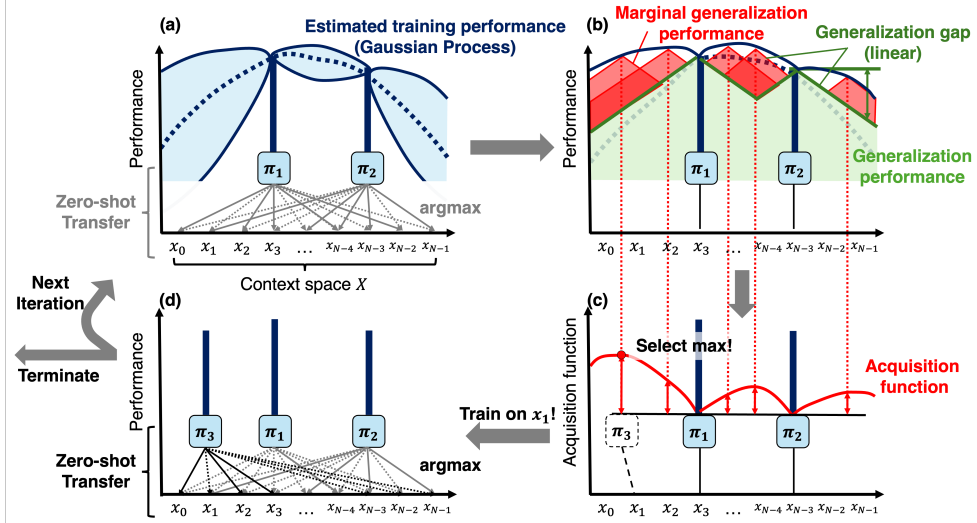
Figure 2: **Overview illustration for Model-based Transfer Learning.** (a) Gaussian process regression is used to estimate the training performance across tasks using existing policies; (b) marginal generalization performance (red area) is calculated using upper confidence bound of estimated training 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.

variable x within a context space X (assumed bounded). The context variable x can influence dynamics, rewards, and initial state distributions [14, 32, 5]. We define source task performance $J(\pi_x, x; \text{Alg})$ as follows: we train a policy π_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 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 π_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 $x, x' \in X$. This experiences performance degradation, also called *generalization gap* [17, 23]. For instance, Figure 3 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. We observe the *generalization performance*, denoted by

Figure 3: 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.

$J(\pi_x, x')$, 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':

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

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*

3

(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 π_{x_k} as π_k. We denote the sequence $x_1, x_2, ..., x_k$ by $x_{1:k}$ and $\pi_1, \pi_2, ..., \pi_k$ 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 π_{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:*

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

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

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

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) 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.

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 π 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 3, we estimate the generalization gap with a linear function of contextual similarity.

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 θ 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, 6]. 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 $V(x_k; \pi_{1:k-1})$

4

(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 $P(\hat{J}_k \mid D_{k-1}) = \mathcal{N}(\mu_k(x), \sigma_k^2(x))$. $\mu_k(x)$ and $\sigma_k^2(x)$ are defined as $\mu_k(x) = \mathbb{E}[\hat{J}(\pi_x,x)] + \mathbf{k}^\top(\mathbf{K} + \sigma^2\mathbf{I})^{-1}\mathbf{y}$ 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 $\mathbf{K} = [k(x_i,x_j)]_{1 \le i,j \le k-1}$. 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) [41]. 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:

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

where $[\cdot]_+$ represents $\max(\cdot, 0)$ and we can use various forms of β_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 $r_k = V(x_k^*; \pi_{1:k-1}) - V(x_k; \pi_{1:k-1})$, where $V(x_k^*; \pi_{1:k-1})$ represents the maximum generalization performance achievable across all tasks, and $V(x_k; \pi_{1:k-1})$ 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. [41], 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. 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 [41].

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 $Pr\left[R_K \le \sqrt{KC_1\beta_K\gamma_K}\right] \ge 1 - \delta$, where $C_1 := \frac{8}{\log(1+\sigma^{-2})} \ge 8\sigma^2$ 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 3, we observe that performance loss decreases as the context similarity increases. We model the degradation from

5

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 3, 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 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 [41] 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 4.
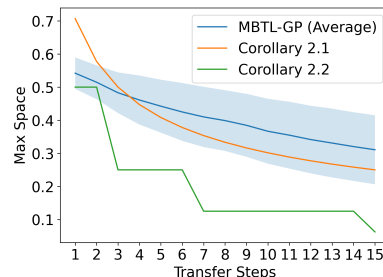
Figure 4: Empirical results of the restriction of search space by MBTL compared to two examples from Corollaries 2.1 and 2.2.

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 [5]. In addition, we study problems from RL for intelligent transportation systems, using [49] 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.

6

Table 1: Comparative performance of different methods on traffic CMDPs ($K = 15$)

Benchmark (CMDP)		Baselines		Multi-policy 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
Traffic Signal	Road Length	**0.9409**	0.8242	0.9366	0.9337	0.9349	0.9364	0.9409
Traffic Signal	Inflow	0.8646	0.8319	**0.8699**	0.8712	0.8682	0.8432	0.8768
Traffic Signal	Speed Limit	0.8857	0.6083	**0.8872**	**0.8872**	**0.8874**	0.8867	0.8876
Eco-Driving	Penetration Rate	0.526	0.1945	0.6212	**0.6278**	0.5992	0.6148	0.666
Eco-Driving	Inflow	0.4061	0.2229	0.5077	**0.5513**	0.5299	0.5172	0.5528
Eco-Driving	Green Phase	0.385	0.4228	0.4724	0.4697	0.4678	**0.4985**	0.5027
AA-Ring-Acc	Hold Duration	0.8362	**0.9209**	0.9306	0.9225	0.9136	**0.9382**	0.9552
AA-Ring-Vel	Hold Duration	0.9589	0.972	**0.9819**	**0.9819**	**0.9819**	0.9818	0.9822
AA-Ramp-Acc	Hold Duration	0.4276	0.5158	0.6750	0.6821	0.6882	**0.6964**	0.7111
AA-Ramp-Vel	Hold Duration	0.5473	0.5034	**0.7170**	**0.7386**	0.6918	**0.7273**	0.7686
Average		0.6778	0.6017	0.7600	0.7666	0.7563	0.7641	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.4.3.
‡AA: Advisory autonomy benchmark, Ring: Single lane ring, Ramp: Highway ramp, Acc: Acceleration guidance, Vel: Speed guidance.

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 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 [36]. For statistical reliability, we run each experiment three times with different random seeds. 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 benchmarks. First, while most traffic lights still operate on fixed schedules, RL can be used to design adaptive (1) **Traffic signal control** to optimize traffic [8, 24]. 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 [11], 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 [13, 47, 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 [40, 7, 15]. 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 5 summarize the results. Notably, the Oracle far outperforms the standard baselines (independent and multi-task training), indicating the potential for transfer learning and intelligent training of multiple models, respectively. MBTL rapidly approaches the Oracle within ≈ 10 transfer steps, indicating that the GP effectively models the training performance and linear generalization gap models the 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 training often (but not always) outperforms multi-task training in our experiments. Yet, our experiments show that MBTL outperforms both independent and multi-task baselines and matches their performance with **up to 30x improved sample efficiency**. Among the multi-policy baselines, MBTL often outperforms the

7
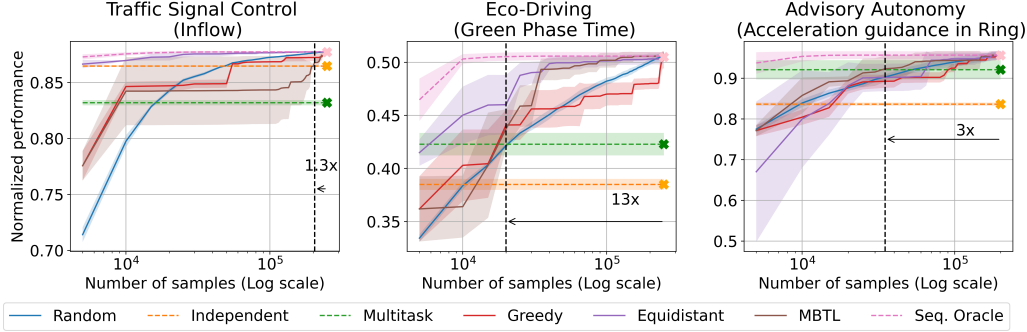
Figure 5: **Traffic CMDP results.** Method comparison of normalized performance over N tasks. MBTL efficiently selects source training tasks. The black dotted line indicates the first training step within MBTL that exceeds both independent and multi-task baselines, with up to 30x fewer samples.

Table 2: Comparative performance of different methods on standard control CMDPs ($K = 15$)

Benchmark (CMDP)		Baselines		Multi-policy 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	0.7607	0.7601	**0.7774**	**0.7755**	0.7969
Pendulum	**Mass**	0.6237	0.5793	0.6647	**0.6794**	**0.6887**	0.6667	0.7132
Pendulum	**Timestep**	0.8135	0.7247	**0.8331**	0.8292	**0.8497**	**0.8333**	0.8801
Cartpole	**Mass of Cart**	**0.9466**	0.7153	0.8961	0.9044	0.8299	0.9356	0.9838
Cartpole	**Length of Pole**	0.9110	0.5441	**0.9497**	0.9484	0.9424	**0.9488**	0.9875
Cartpole	**Mass of Pole**	0.9560	0.6073	0.9870	**0.9927**	**0.9916**	0.9813	1.0000
BipedalWalker	**Gravity**	0.9281	0.7898	**0.9654**	**0.9666**	**0.9656**	**0.9655**	0.9674
BipedalWalker	**Friction**	0.9317	0.9051	**0.9739**	**0.9738**	**0.9738**	0.9725	0.9778
BipedalWalker	**Scale**	0.8694	0.7452	0.8910	**0.8975**	**0.8990**	0.8962	0.9107
HalfCheetah	**Gravity**	0.6679	0.6292	0.9086	0.9000	0.9089	**0.9214**	0.9544
HalfCheetah	**Friction**	0.6693	0.7242	0.9314	**0.9457**	0.9184	0.9225	0.9663
HalfCheetah	**Stiffness**	0.6561	0.7007	0.9191	0.9097	**0.9295**	**0.9205**	0.9674
Average		0.8093	0.6957	0.8901	0.8923	0.8896	0.8950	0.9255

†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.4.3.

heuristic multi-policy baselines, indicating the value of adaptively selecting source tasks based on feedback. The multi-policy baselines, such as random, equidistant, and greedy strategy, also generally outperform Independent and Multi-task, 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
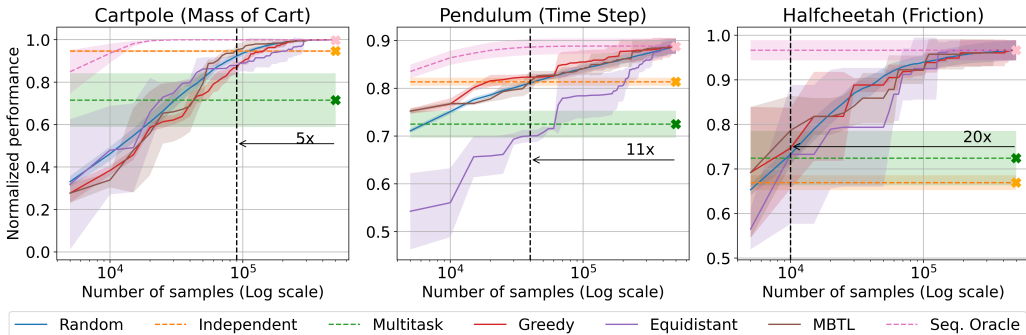
Figure 6: **Continuous control CMDP results.** Method comparison of normalized performance over N tasks. The black dotted line indicates the first training step within MBTL that exceeds both independent and multi-task baselines, with up to 43x improved sample efficiency.

8
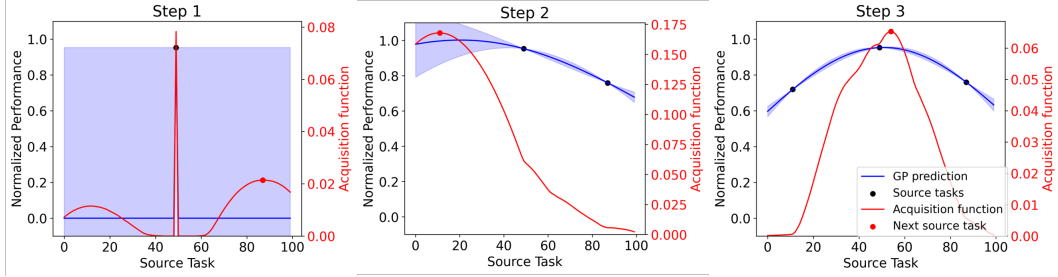
Figure 7: The GP sequentially updates estimates of the performance function (blue) based on previously trained models. Then, MBTL selects the next source task that maximizes the acquisition function (red). (CMDP: Pendulum (Time step)).
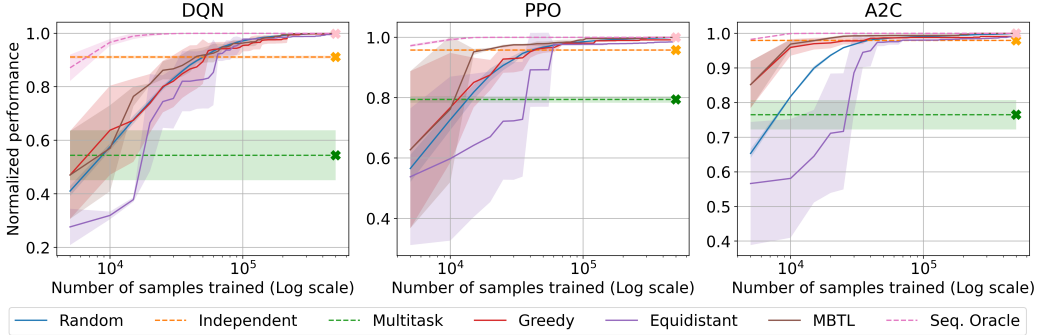
Figure 8: Sensitivity analysis on the DRL algorithm underlying MBTL (DQN, PPO, and A2C), tested on Cartpole with varying length of pole. MBTL remains effective.

To probe the generality of MBTL, we utilize context-extended versions of standard RL environments from CARL benchmark library [5] to evaluate our methods under varied contexts. For the Cartpole 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 experimental details in Appendix A.4.

Results. The results summarized in Table 2 demonstrate sample efficiency and competitive performance of multi-policy training including MBTL across diverse control domains, often closely trailing the Oracle only with a small number of trained policies. Figure 6 shows that with the exception of a few context variations, MBTL generally shows superior performance. Specifically, Figure 7 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 **5-43x** fewer samples, highlighting its effectiveness in reducing training requirements.

5.3.1 Sensitivity analysis

DRL algorithms. Figure 8 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 9 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
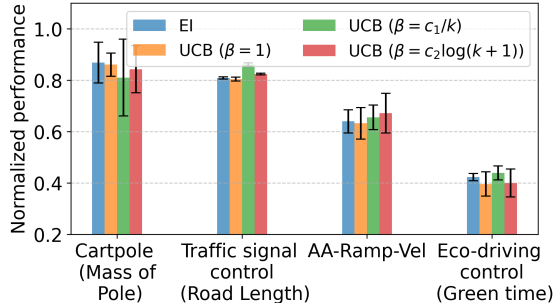
Figure 9: Sensitivity analysis on acquisition functions.

9

variance for balancing exploration and exploitation. 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, which utilizes side information about the problem variations to improve the generalization and robustness. In particular, CRL formalizes generalization in DRL using CMDPs [14, 32, 5], which incorporate context-dependent dynamics, rewards, and initial state distributions into the formalism of MDPs. The contexts of CMDPs are not always visible during training [23]. When they are visible, they can be directly used as side information by conditioning the policy on them [39]. In this paper, we focus on a scenario where the learner can choose which context-MDP to train on. This contrasts with other CRL works that assume context-MDPs arrive from a fixed distribution.

Multi-task training. Multi-task methods can help address CRL by exploiting shared structure across tasks. Prior work has leveraged techniques such as policy sketches for task decomposition [2] and distilled policies that capture common behaviors [46]. However, a key limitation arises when the context is unobserved, effectively transforming the CMDP into a partially observable setting [23, 9], which complicates multi-task training. Another challenge is negative transfer, wherein training on tasks that are too dissimilar leads to instability or outright failure [22, 42, 44]. Although more recent multi-task approaches such as MOORE [16] and PaCo [43] have shown promise, they often focus on discrete task sets and are thus less suited to CRL, where tasks span a broad continuum of contexts. In this work, we include multi-task learning as a baseline to benchmark our methods.

Zero-shot transfer and policy reuse. Zero-shot transfer—where models trained for one environment directly perform in new, unseen settings without additional training [23]—is an important strategy in CRL settings. For solving CMDP problems, prior works attempted to utilize zero-shot transfer to solve CMDP problems by approximation on RL algorithm and hypernetworks that maps from parameterized CMDP to a family of near-optimal solutions [35]. Sinapov et al. [38] 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. Taken together, these approaches highlight the importance of policy reuse, where efficiently selecting or adjusting a pre-trained policy accelerates learning and improves robustness in new contexts.

Source task selection. In the context of transfer learning, selecting appropriate source tasks is crucial. Li and Zhang [25] 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. Building upon these insights, we formulate the source task selection problem for CRL, enabling zero-shot transfer by estimating training performance online and leveraging structural generalization across context variations.

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 43x 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 with a reliance on the explicit similarity of context variables. Promising directions of future work include studying high-dimensional context spaces and formalizing task similarity, as well as the development of new real-world CMDP benchmarks.

10

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] 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.

[6] 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.

[7] 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.

[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.

11

[15] 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.

[16] Ahmed Hendawy, Jan Peters, and Carlo D'Eramo. Multi-task reinforcement learning with mixture of orthogonal experts. *arXiv preprint arXiv:2311.11385*, 2023.

[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] Vindula Jayawardana, Baptiste Freydt, Ao Qu, Cameron Hickert, Zhongxia Yan, and Cathy Wu. Intersectionzoo: Eco-driving for benchmarking multi-agent contextual reinforcement learning. *arXiv preprint arXiv:2410.15221*, 2024.

[22] 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.

[23] 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.

[24] 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.

[25] 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.

[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.

12

[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] Antonin Raffin, Ashley Hill, Adam Gleave, Anssi Kanervisto, Maximilian Ernestus, and Noah Dormann. Stable-baselines3: Reliable reinforcement learning implementations. *Journal of Machine Learning Research*, 22(268):1–8, 2021. URL http://jmlr.org/papers/v22/20-1364.html.

[35] 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.

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

[37] 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.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] Lingfeng Sun, Haichao Zhang, Wei Xu, and Masayoshi Tomizuka. Paco: Parameter-compositional multi-task reinforcement learning. *Advances in Neural Information Processing Systems*, 35:21495–21507, 2022.

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] Christopher KI Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*, volume 2. MIT press Cambridge, MA, 2006.

13

[49] 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.

14

A Appendix

Contents

A.1 Notation 16

A.2 Model-Based Transfer Learning (MBTL) Algorithm 17

A.3 Theoretical analysis 17

    A.3.1 Proof of Theorem 2 17

    A.3.2 Proof of Corollary 2.1 18

    A.3.3 Proof of Corollary 2.2 18

A.4 Experiment details 19

    A.4.1 Details about Gaussian process (GP) Regresstion 19

    A.4.2 Accuracy of generalization gap assumption 19

    A.4.3 Results of table with standard deviation 20

    A.4.4 Detailed sample complexity comparison results 21

    A.4.5 Details about traffic signal control benchmark 21

    A.4.6 Details about eco-driving control benchmark 24

    A.4.7 Details about advisory autonomy benchmark 25

    A.4.8 Details of control benchmarks 26

    A.4.9 Details about Cartpole benchmark 27

    A.4.10 Details about Pendulum benchmark 28

    A.4.11 Details about BipedalWalker benchmark 30

    A.4.12 Details about HalfCheetah benchmark 31

    A.4.13 Implementation of the recent multi-task baselines 33

A.5 Potential impacts 34

15

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 ($x \in X$)
x'	Target task ($x' \in X$)
π_x	Trained policy from source task ($x \in X$)
x_k	Selected source task at transfer step k ($k = 1, ..., K$)
$\mathcal{M}_x$	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 10 helps understand the discrepancy between the observed generalized performance and the predicted one. Figure 11 illustrates how to calculate the marginal improvement of expected generalized performance ($\hat{V}(x; \pi_{1:k-1}) - V(x_{1:k-1})$).
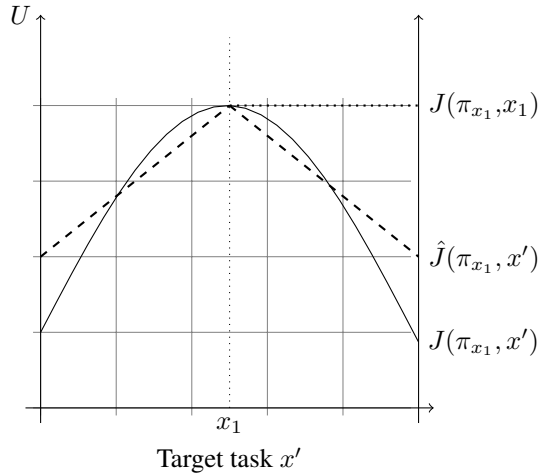
Figure 10: Illustration of the discrepancy between observed (J) and predicted ($\hat{J}$) generalized performance after training on source task x_1 and attempting zero-shot transfer to x'.

16
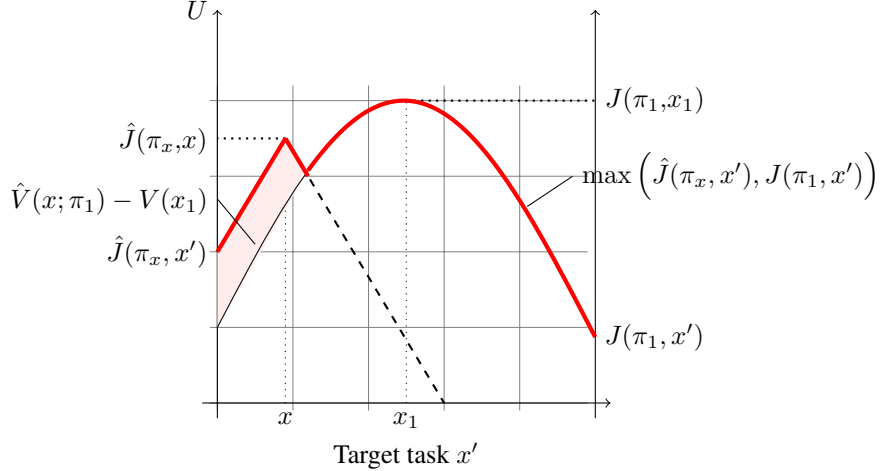
Figure 11: Step for choosing x_2 that maximizes the estimated marginal improvement ($\hat{V}(x; \pi_1) - V(x_1)$). $\hat{V}(x; \pi_1)$ corresponds to the red area under the red line and $V(x_1)$ as the area under $J(\pi_1, x')$.

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* : $J, V = 0 \; \forall x \in X$, $\boldsymbol{\pi} = \{\}$, $k = 1$
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:    $\pi_k \leftarrow \textbf{Train}(\mathcal{M}_{x_k})$
11:    $\boldsymbol{\pi} \leftarrow \boldsymbol{\pi} \cup \{\pi_k\}$
12:    $k \leftarrow k + 1$
13: **end while**
14: Zero-shot transfer and calculate generalization performance $V(x_1, ..., x_k)$
15: **Output:** Set of policies $\boldsymbol{\pi}$ 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 [41].

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 $2|X_k|\beta_k^{1/2}\sigma_{k-1}(x)/|X|$.*

17

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

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

$\square$

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:

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

The cumulative regret can be bounded as below:

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

$\square$

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:

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

The cumulative regret can be bounded as below:

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

$\square$

18

A.4 Experiment details

A.4.1 Details about Gaussian process (GP) Regresstion

We use the `GaussianProcessRegressor` implementation from `scikit-learn`, which follows Algorithm 2.1 of [48]. Specifically, we construct a kernel by multiplying a constant kernel

$$C(\theta) = 1.0, \quad \theta \in (10^{-3}, 10^3),$$

by a radial basis function (RBF) kernel

$$k_{\mathrm{RBF}}(\mathbf{x}, \mathbf{x}'; \ell) = \exp\Big(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\ell^2}\Big)$$

with an initial length scale $\ell = 1.0$ (constrained to lie in the range $[10^{-2}, 10^2]$). To determine the hyperparameters, we begin by generating synthetic data that aligns with our modeling assumptions, including constant training performance and a linear generalization gap, while introducing noise to degrade generalization performance by up to 10%, sampled from a uniform distribution. We vary the GP hyperparameters, including noise standard deviation over the set $\{0.001, 0.01, 0.1, 1\}$, the number of restarts for the optimizer over $\{5, 6, \dots, 15\}$, and explore several kernel configurations on the synthetic data. We then select the hyperparameter configuration that maximizes the average predictive performance. Specifically, we choose a noise standard deviation of $\sigma = 0.001$ and perform 15 random restarts of the hyperparameter optimizer to reduce the risk of convergence to poor local minima. We use the same GP hyperparameter configuration across all experiments and benchmarks.

A.4.2 Accuracy of generalization gap assumption

In Figure 12, we report the Pearson correlation between the observed generalization gap and the estimated gap under our linear assumption (Assumption 1). Each histogram shows how strongly the two measures align across various tasks in standard control (blue) and traffic (red) benchmarks. Many tasks cluster around moderate positive correlations (0.3–0.5), suggesting that a linear function of context similarity can reasonably capture the gap in most scenarios. However, certain tasks—such as Eco-driving—exhibit higher correlations (above 0.6), whereas others—such as HalfCheetah—are closer to 0, indicating that the assumption holds more effectively in some domains than in others.
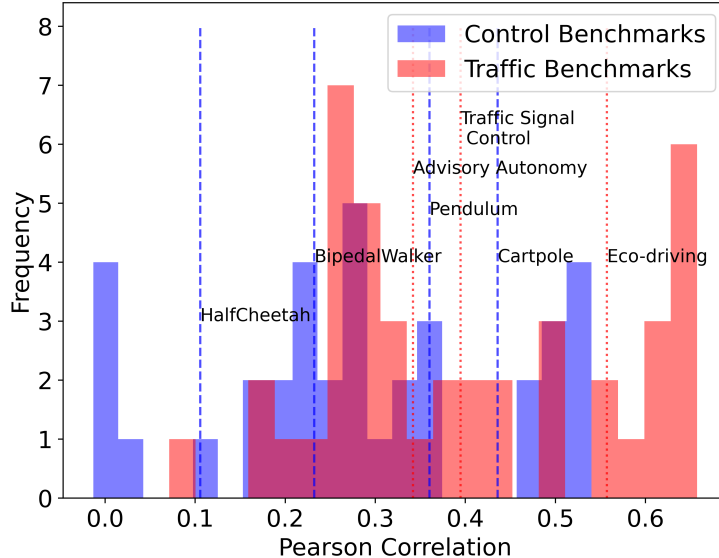
Figure 12: **Accuracy of linear generalization gap assumption.** Pearson correlation analysis on the observed generalization gap and the estimated gap under our linear assumption.

19

A.4.3 Results of table with standard deviation

Table 4: Comparative performance of different methods on context-variant traffic and benchmark CMDPs ($K = 15$)

Benchmark (CMDP)		Baselines		Multi-policy 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
Traffic Signal	Road Length	**0.9409** (0.0002)	0.8242 (0.0538)	0.9366 (0.0009)	0.9337 (0.0044)	0.9349 (0.0021)	0.9364 (0.0059)	0.9409 (0.0006)
Traffic Signal	Inflow	0.8646 (0.0009)	0.8319 (0.0040)	**0.8699** (0.0011)	**0.8712** (0.0044)	**0.8682** (0.0008)	0.8432 (0.0372)	0.8768 (0.0012)
Traffic Signal	Speed Limit	0.8857 (0.0005)	0.6083 (0.0403)	**0.8872** (0.0002)	**0.8872** (0.0004)	**0.8874** (0.0004)	0.8867 (0.0002)	0.8876 (0.0002)
Eco-Driving	Penetration Rate	0.5260 (0.0087)	0.1945 (0.0057)	**0.6212** (0.0041)	**0.6278** (0.0332)	0.5992 (0.0007)	**0.6148** (0.0145)	0.6660 (0.0049)
Eco-Driving	Inflow	0.4061 (0.0094)	0.2229 (0.0010)	0.5077 (0.0114)	**0.5513** (0.0113)	0.5299 (0.0456)	0.5172 (0.0415)	0.5528 (0.0096)
Eco-Driving	Green Phase	0.3850 (0.0063)	0.4228 (0.0184)	0.4724 (0.0069)	0.4697 (0.0318)	0.4678 (0.0147)	**0.4985** (0.0096)	0.5027 (0.0058)
AA-Ring-Acc	Hold Duration	0.8362 (0.0048)	**0.9209** (0.0235)	**0.9306** (0.0087)	**0.9225** (0.0252)	0.9136 (0.0193)	**0.9382** (0.0216)	0.9552 (0.0132)
AA-Ring-Vel	Hold Duration	0.9589 (0.0096)	0.9720 (0.0035)	**0.9819** (0.0001)	**0.9819** (0.0003)	**0.9819** (0.0001)	**0.9818** (0.0002)	0.9822 (0.0000)
AA-Ramp-Acc	Hold Duration	0.4276 (0.0066)	0.5158 (0.045)	**0.6750** (0.0275)	0.6821 (0.0809)	0.6882 (0.0715)	**0.6964** (0.0522)	0.7111 (0.0479)
AA-Ramp-Vel	Hold Duration	0.5473 (0.0222)	0.5034 (0.014)	**0.7170** (0.0384)	**0.7386** (0.0410)	0.6918 (0.084)	**0.7273** (0.0636)	0.7686 (0.0581)
Pendulum	Length	0.7383 (0.0034)	0.6830 (0.0008)	0.7607 (0.0072)	0.7601 (0.0074)	**0.7774** (0.0041)	**0.7755** (0.0215)	0.7969 (0.0123)
Pendulum	Mass	0.6237 (0.0023)	0.5793 (0.0041)	0.6647 (0.0065)	**0.6794** (0.0376)	**0.6887** (0.0116)	0.6667 (0.013)	0.7132 (0.0129)
Pendulum	Timestep	0.8135 (0.0103)	0.7247 (0.0488)	**0.8331** (0.0084)	0.8292 (0.0189)	**0.8497** (0.0322)	**0.8333** (0.0130)	0.8801 (0.0198)
Cartpole	Mass of Cart	**0.9466** (0.0065)	0.7153 (0.2195)	0.8961 (0.0214)	0.9044 (0.055)	0.8299 (0.0392)	0.9356 (0.0239)	0.9838 (0.0131)
Cartpole	Length of Pole	0.9110 (0.0065)	0.5441 (0.1614)	**0.9497** (0.0044)	**0.9484** (0.042)	0.9424 (0.031)	**0.9488** (0.0145)	0.9875 (0.0047)
Cartpole	Mass of Pole	0.9560 (0.0128)	0.6073 (0.0948)	0.9870 (0.0050)	**0.9927** (0.0014)	**0.9916** (0.0030)	0.9813 (0.0103)	1.0000 (0.0000)
BipedalWalker	Gravity	0.9281 (0.0034)	0.7898 (0.0928)	**0.9654** (0.0004)	**0.9666** (0.0022)	**0.9656** (0.0021)	**0.9655** (0.001)	0.9674 (0.0014)
BipedalWalker	Friction	0.9317 (0.0074)	0.9051 (0.0734)	**0.9739** (0.0003)	**0.9738** (0.0022)	**0.9738** (0.0013)	0.9725 (0.0038)	0.9778 (0.0013)
BipedalWalker	Scale	0.8694 (0.0087)	0.7452 (0.0938)	**0.8910** (0.0079)	**0.8975** (0.0118)	**0.8990** (0.0135)	**0.8962** (0.0123)	0.9107 (0.004)
HalfCheetah	Gravity	0.6679 (0.0162)	0.6292 (0.0258)	**0.9086** (0.0078)	0.9000 (0.0600)	**0.9089** (0.0235)	**0.9214** (0.0232)	0.9544 (0.0221)
HalfCheetah	Friction	0.6693 (0.0203)	0.7242 (0.1056)	**0.9314** (0.0175)	**0.9457** (0.0295)	0.9184 (0.0184)	**0.9225** (0.0392)	0.9663 (0.0276)
HalfCheetah	Stiffness	0.6561 (0.0101)	0.7007 (0.1126)	**0.9191** (0.01)	0.9097 (0.027)	**0.9295** (0.0169)	**0.9205** (0.0161)	0.9674 (0.0286)
Average		0.7775 (0.197)	0.6337 (0.2395)	0.8309 (0.0089)	0.8352 (0.024)	0.829 (0.0198)	0.8355 (0.0199)	0.8833 (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.

20

A.4.4 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 represent the sample efficiency of MBTL.

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

Task	Context Variation	k^*	N
Pendulum	Length	[4, 5, 5]	100
Pendulum	Mass	[3, 3, 3]	100
Pendulum	Timestep	[8, 5, 14]	100
Cartpole	Mass of Cart	[15, 25, 18]	100
Cartpole	Length of Pole	[10, 6, 14]	100
Cartpole	Mass of Pole	[9, 9, 12]	100
BipedalWalker	Gravity	[2, 3, 4]	100
BipedalWalker	Friction	[3, 2, 2]	100
BipedalWalker	Scale	[7, 6, 1]	100
HalfCheetah	Gravity	[10, 2, 5]	100
HalfCheetah	Friction	[1, 2, 13]	100
HalfCheetah	Stiffness	[1, 2, 5]	100
AA-Ring-Acc	Hold Duration	[5, 7, 30]	40
AA-Ring-Vel	Hold Duration	[2, 1, 1]	40
AA-Ramp-Acc	Hold Duration	[1, 2, 2]	40
AA-Ramp-Vel	Hold Duration	[6, 4, 4]	40
Traffic Signal	Road Length	[50, 50, 11]	50
Traffic Signal	Inflow	[2, 44, 41]	50
Traffic Signal	Speed Limit	[10, 4, 8]	50
Eco-Driving	Penetration Rate	[3, 5, 1]	50
Eco-Driving	Inflow	[2, 1, 1]	50
Eco-Driving	Green Phase	[3, 3, 5]	50

A.4.5 Details about traffic signal control benchmark

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, 24], though challenges persist in generalizing across various intersection configurations [19].

Figure 13 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. To represent various real-world conditions, we vary factors such as road length, inflow rate, and speed limits from 0.1 to 5 times; by default, the road length is 750 meters, the flow rate is 1000 vehicles per hour, and the speed limit is 13.89 m/s.

21
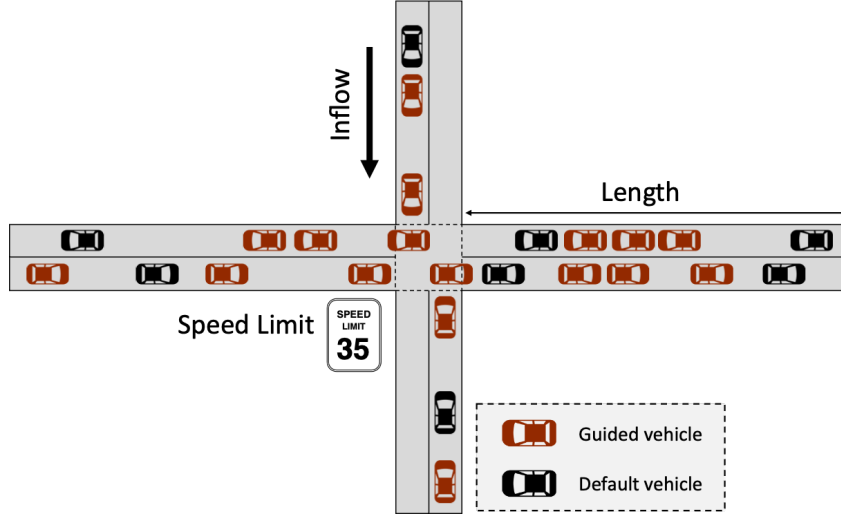
Figure 13: Illustration of the traffic networks in traffic signal control task.

Training configuration   We used the microscopic traffic simulation called Simulation of Urban MObility (SUMO) [26] v.1.16.0 and PPO for RL algorithm [36]. We utilized the default implementation of the PPO algorithm with default hyperparameters provided by the Stable-Baselines3 library [34]. All experiments are done on a distributed computing cluster equipped with 48 Intel Xeon Platinum 8260 CPUs.

License   Traffic signal control benchmark falls under MIT License.
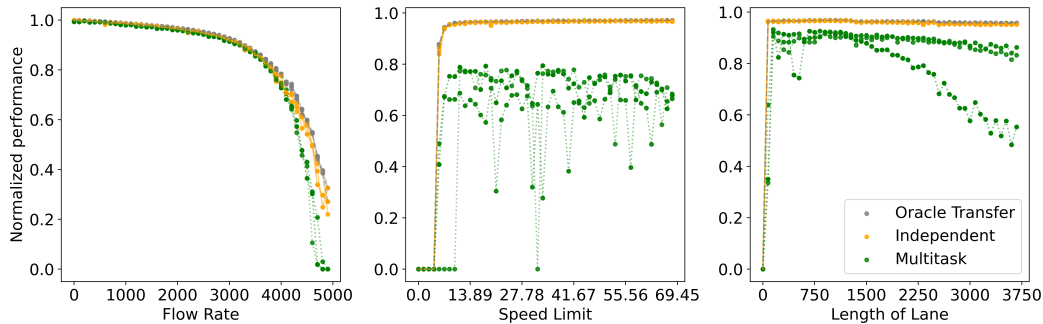
Figure 14: Normalized performance of three DRL-based methods—Oracle Transfer (gray), independent training (orange), and multi-task training (green)—under different traffic-signal benchmarks: flow rate (left), speed limit (middle), and lane length (right). While independent and multi-task training approaches exhibit higher variance and reduced asymptotic performance, Oracle Transfer benefits from zero-shot transfer with full information, demonstrating more stable and generally higher performance.

Potential of multi-policy training and zero-shot transfer   Figure 14 shows how each approach adapts to variations in flow rate, speed limit, and lane length for a traffic signal control benchmark. The y-axis shows normalized performance, with higher values indicating better control policies. Oracle Transfer consistently achieves superior performance across these different settings, owing to its ability to leverage full task information in a zero-shot transfer manner. By contrast, independent and multi-task training exhibit more pronounced performance drops and greater instability when faced with shifts in problem parameters, underscoring the challenges of generalizing policies in traditional DRL approaches.

22

Transferability heatmap   Figure 15 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
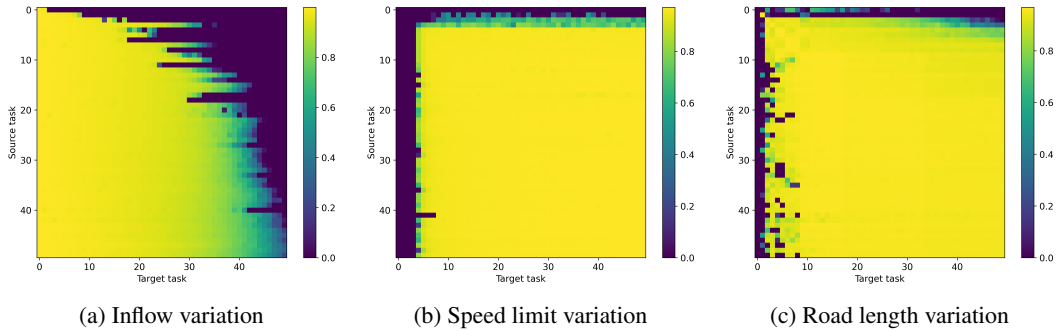
Figure 15: Examples of transferability heatmap for traffic signal control.

Results   Figure 16 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 multi-task 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.
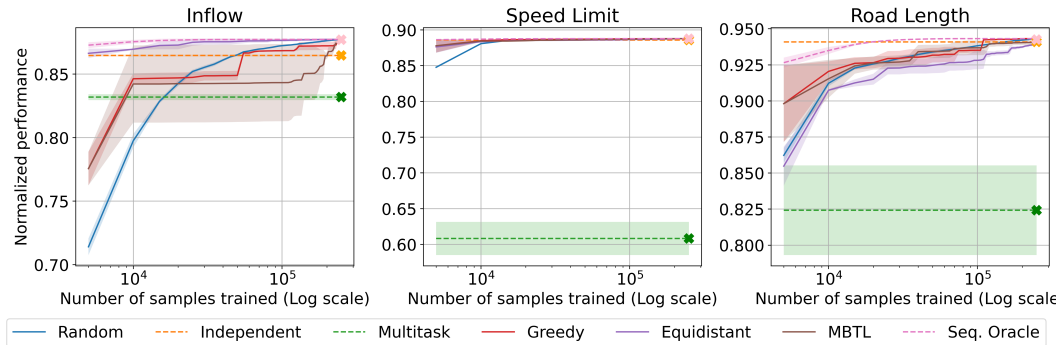
Figure 16: Comparison of normalized generalized performance of all target tasks: Traffic signal control.

23

A.4.6 Details about eco-driving control benchmark

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, 47, 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 17 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.
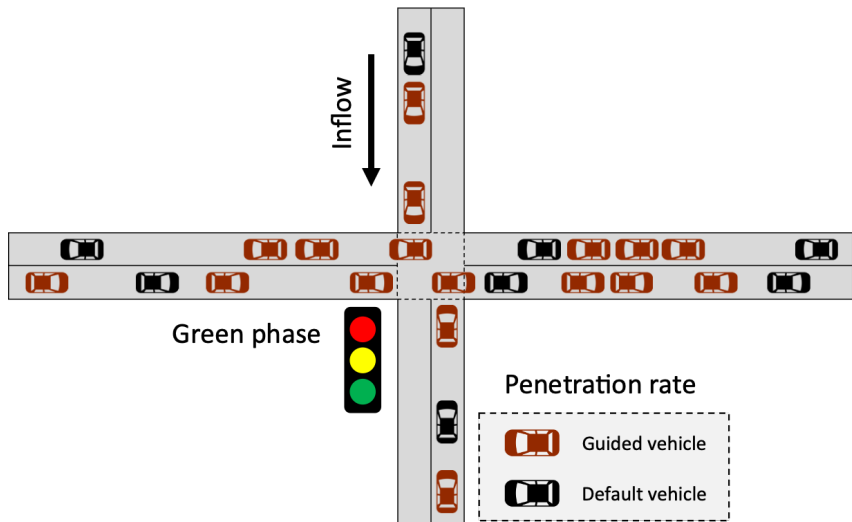
Figure 17: Illustration of the traffic networks in eco-driving control task.

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 [36]. For detailed experimental details and RL hyperparameter configurations, please refer to [20, 21].

License Eco-driving benchmark falls under MIT License [21].

Potential of multi-policy training and zero-shot transfer Figure 1 shows how each RL training paradigm adapts to variations in green phase time, penetration rate, and inflow rate in the eco-driving control benchmark. Oracle Transfer remains the strongest method across all configurations, benefiting from zero-shot transfer. independent training shows unstable performance across different task variations, performance, while multi-task training lags behind. Overall, the trends highlight the advantage of leveraging zero-shot transfer in traffic CMDPs.

Transferability heatmap Figure 18 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.

24

(a) Green phase variation   (b) Inflow variation   (c) Penetration rate variation
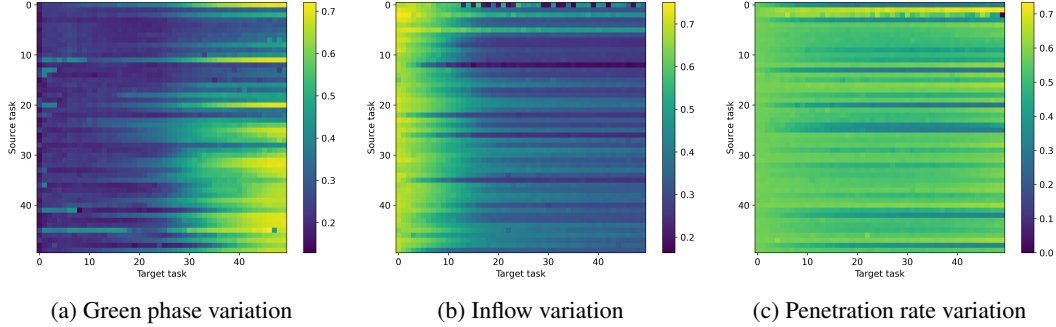
Figure 18: Examples of transferability heatmap for eco-driving control.

Results Figure 19 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.
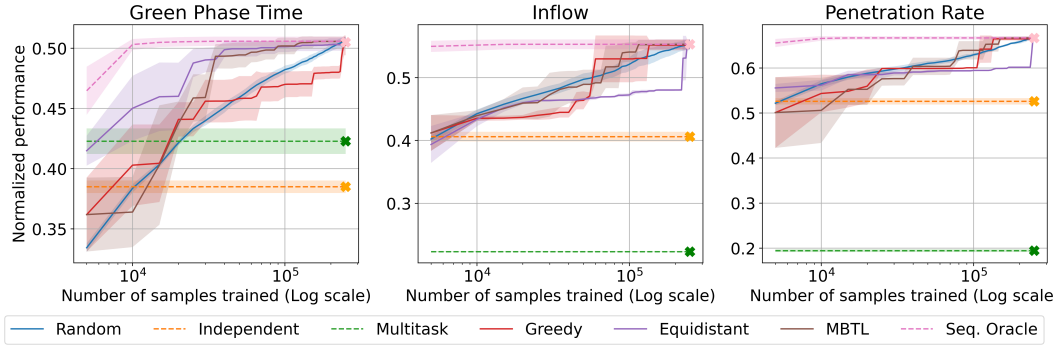
Figure 19: Comparison of normalized generalized performance of all target tasks: Eco-driving control.

A.4.7 Details about advisory autonomy benchmark

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 [40, 7, 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 20 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.

25

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.

License   Advisory autonomy benchmark falls under MIT License [40].
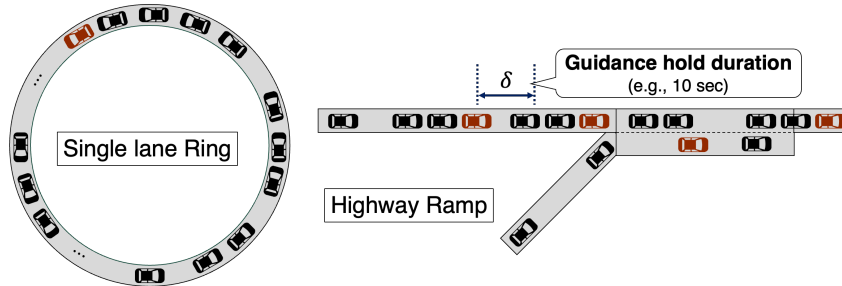
Figure 20: Illustration of the traffic networks in advisory autonomy task.
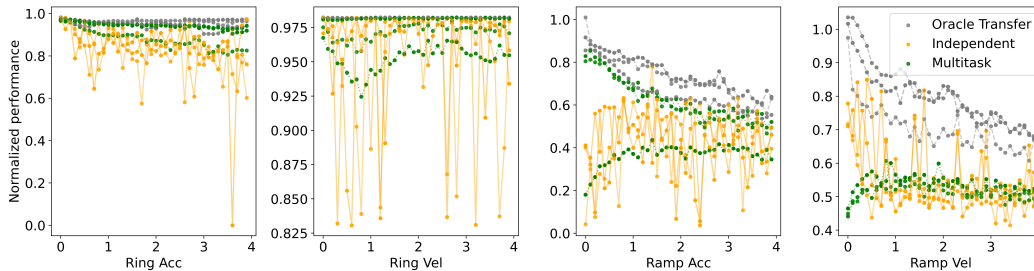
Figure 21: Normalized performance of Oracle Transfer, independent training, and multi-task training under Advisory Autonomy benchmark with human compatibility task variations.

Potential of multi-policy training and zero-shot transfer   Figure 21 shows that ring-road networks tend to yield higher performance and smaller gaps compared to highway ramp scenarios. In addition, independent training exhibits greater performance drop and variability due to training instability. Oracle Transfer retains clear potential improvements over other baselines.

Transferability heatmap   Figure 22 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.

Results   Figure 23 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.

A.4.8   Details of control benchmarks

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.

26

(a) Ring with acceleration guide
(b) Ring with speed guide
(c) Ramp with acceleration guide
(d) Ramp with speed guide
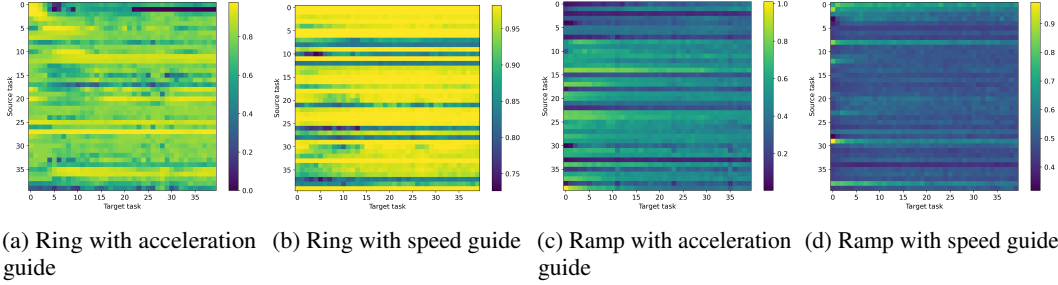
Figure 22: Examples of transferability heatmap for advisory autonomy.
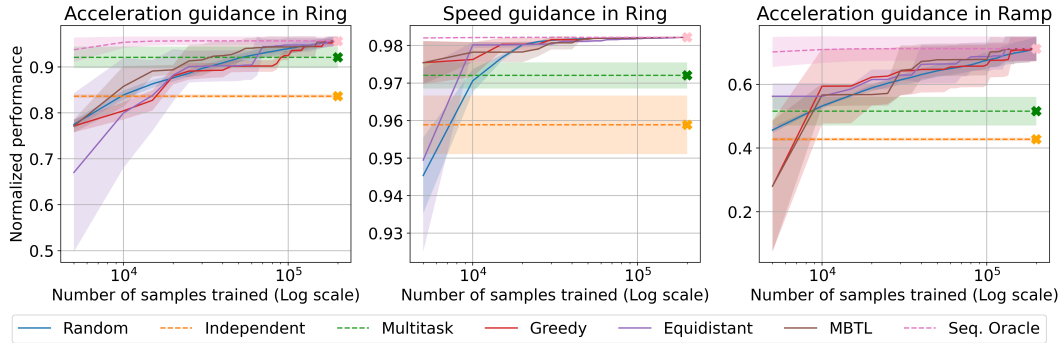
Figure 23: Comparison of normalized generalized performance of all target tasks: Advisory autonomy.

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.

We utilized the default implementation of the DQN, A2C, and PPO algorithm with default hyperparameters provided by the Stable-Baselines3 library [34].

License: CARL falls under the Apache License 2.0 as is permitted by all work that we use [5].

A.4.9 Details about Cartpole benchmark

Potential of multi-policy training and zero-shot transfer Cartpole may be considered a simpler benchmark than traffic benchmarks, yet independent and multi-task training methods still face notable difficulty when faced with context variations. Figure 24 shows that Oracle Transfer performs at the highest performance across problem variations, while independent training or multi-task training shows a larger variance in performance.

27
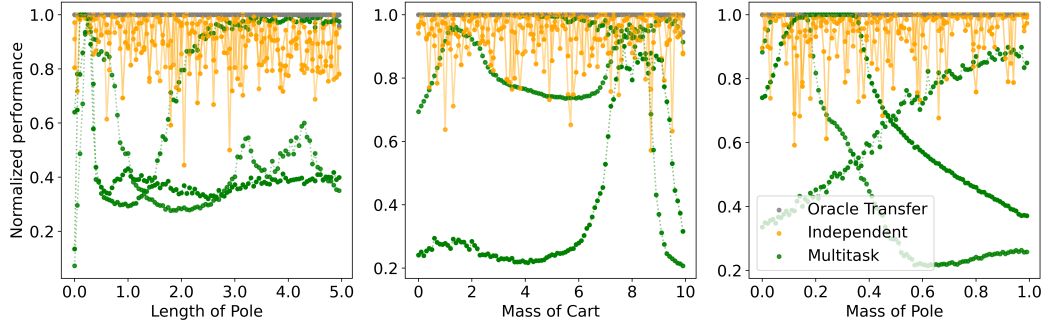
Figure 24: Normalized performance of Oracle Transfer, independent training, and multi-task training in Cartpole benchmarks.

Transferability heatmap Figure 25 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
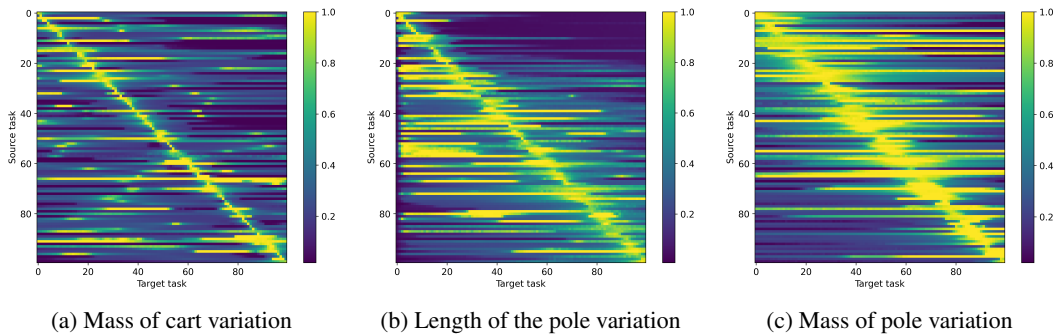
Figure 25: Examples of transferability heatmap for Cartpole.

Results Figure 26 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, performance generally increases with transfer steps, with MBTL strategies achieving the highest scores. This indicates 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.

A.4.10 Details about Pendulum benchmark

Potential of multi-policy training and zero-shot transfer Pendulum is also one of the simplest benchmarks in classic control. In the Pendulum benchmark, we vary three key parameters: time step (left), pole length (middle), and ball mass (right). As shown in Figure 27, a few well-trained policies in specific contexts transfer effectively to new tasks, particularly under Oracle Transfer. For certain configurations (e.g., shorter poles, lighter balls), Independent and Oracle Transfer both excel, while multi-task struggles. These results suggest a remaining performance gap that multi-policy training and zero-shot transfer could help bridge.

28
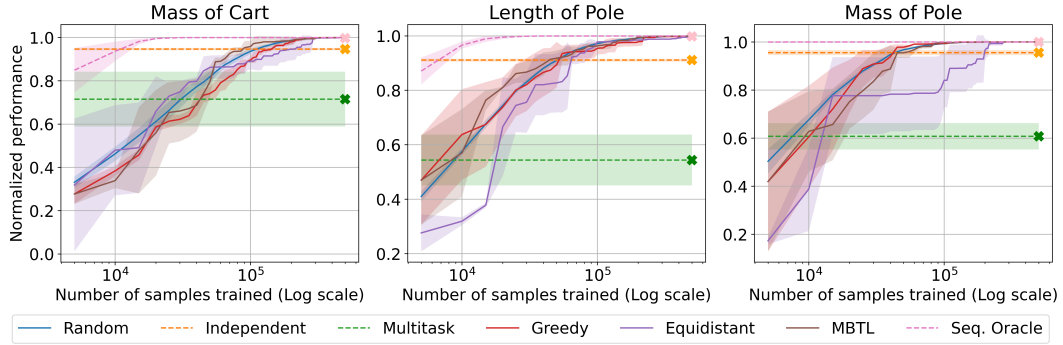
Figure 26: Comparison of normalized generalized performance of all target tasks: Cartpole.
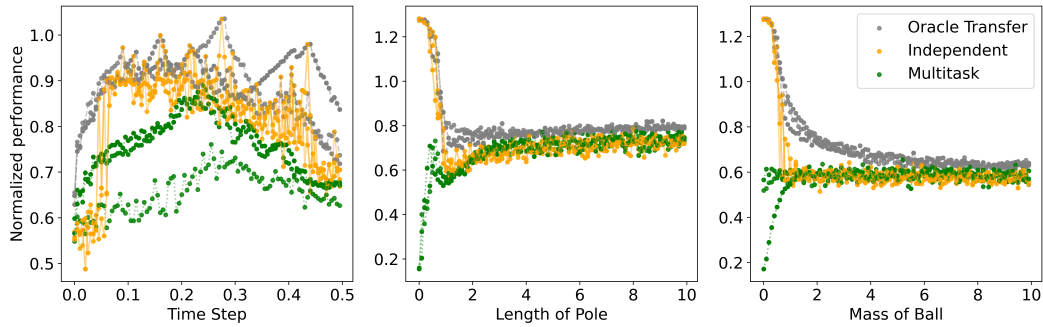
Figure 27: Normalized performance of Oracle Transfer, independent training, and multi-task training in Pendulum benchmarks.

Figure 28 provides a visual mapping of which trained policy is being applied for each specific context in the Pendulum benchmark. CMDP with time step variation demonstrates that the tasks are covered by a few "good" policies nearby. In addition, contexts with shorter poles and lighter balls often gravitate toward a single high-performing policy, whereas more challenging configurations may require a specialized or distinct policy. This illustrates how Oracle Transfer can seamlessly select from a suite of learned policies, demonstrating robust zero-shot transfer and stronger adaptability.
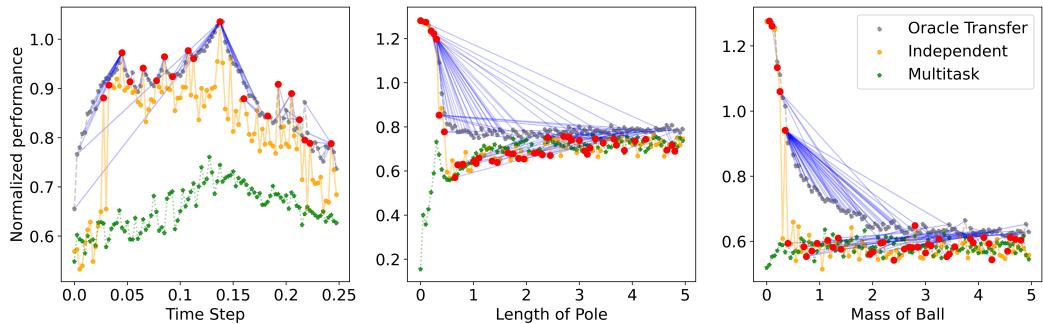
Figure 28: Visulazation on which policy is used to solve specific context-MDP in Pendulum CMDP.

Transferability heatmap   Figure 29 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), transfer-

29

ability 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 varia-   (c) Mass of pendulum variation
                                   tion
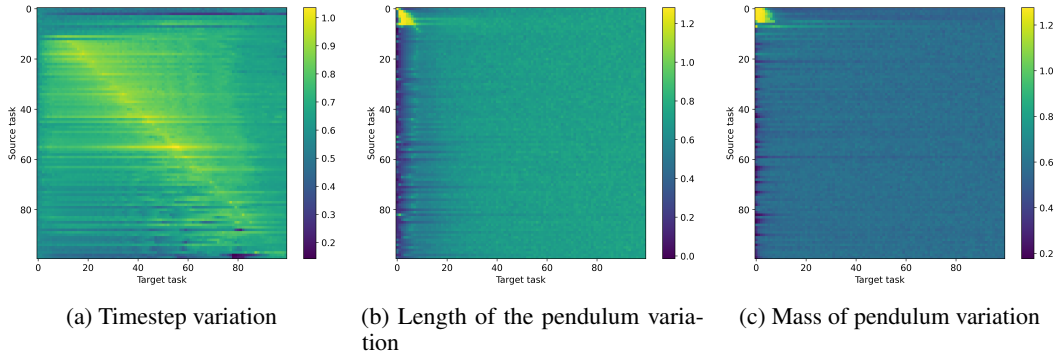
Figure 29: Examples of transferability heatmap for Pendulum.

Results   Figure 30 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.
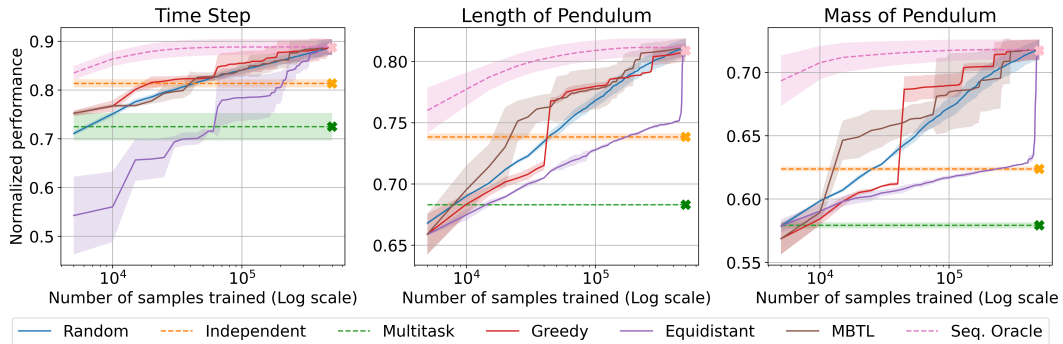
Figure 30: Comparison of normalized generalized performance of all target tasks: Pendulum.

A.4.11   Details about BipedalWalker benchmark

Potential of multi-policy training and zero-shot transfer   Figure 31 compares the performance of different RL training methods in the BipedalWalker benchmark. independent training typically performs nearly as well but suffers intermittent dips. Similarly, multi-task training experiences larger swings, occasionally collapsing to low performance in certain parameter regions. However, Oracle Transfer remains near-perfect across every setting. These patterns highlight how multi-policy training with zero-shot transfer and per-task training generally fare better than a single universal model when faced with diverse environment dynamics.

30
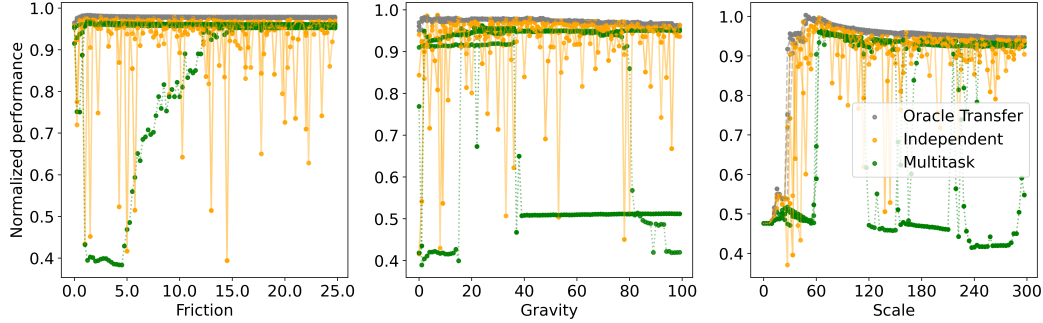
Figure 31: Normalized performance of Oracle Transfer, independent training, and multi-task training in BipedalWalker benchmarks.

Transferability heatmap Figure 32 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
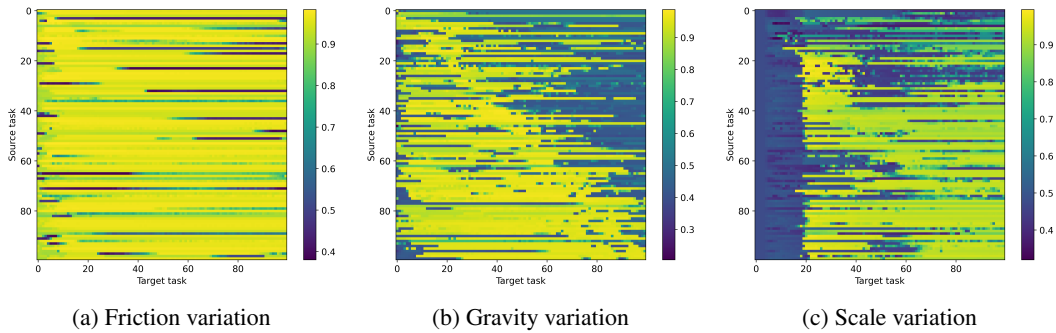
Figure 32: Examples of transferability heatmap for BipedalWalker.

Results Figure 33 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 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.

A.4.12 Details about HalfCheetah benchmark

Potential of multi-policy training and zero-shot transfer In this HalfCheetah benchmark, each subplot examines how policies adapt to changing friction, gravity, and stiffness (Figure 34). Oracle Transfer maintains nearly perfect scores for all parameter ranges, indicating robust zero-shot adaptability. In contrast, independent training experiences larger fluctuations, while multi-task training remains consistent yet at a lower performance plateau. The clear gap between Oracle Transfer and the other methods highlights the advantage of leveraging specialized multi-policy training solutions that effectively transfer across diverse dynamics.

31
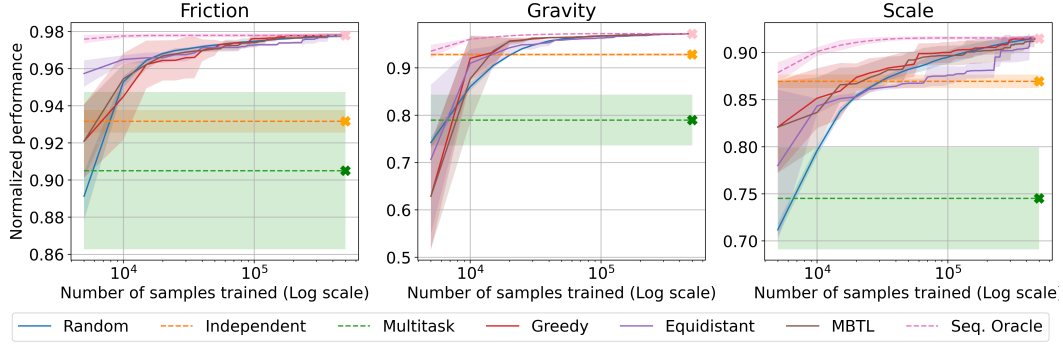
Figure 33: Comparison of normalized generalized performance of all target tasks: BipedalWalker.
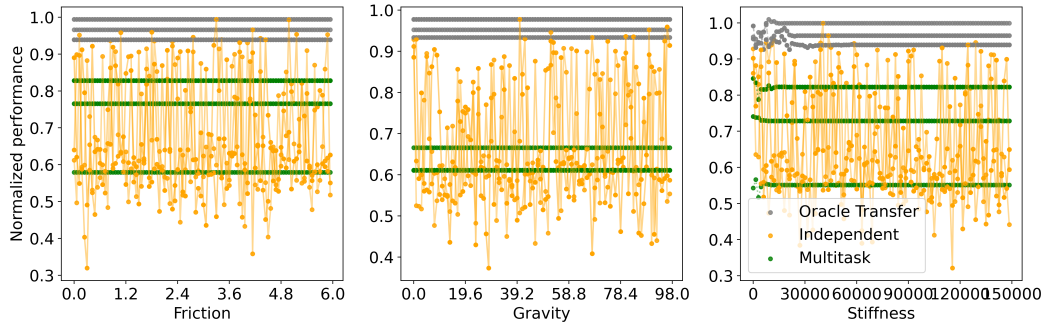
Figure 34: Normalized performance of Oracle Transfer, independent training, and multi-task training in HalfCheetah benchmarks.

Transferability heatmap Figure 35 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
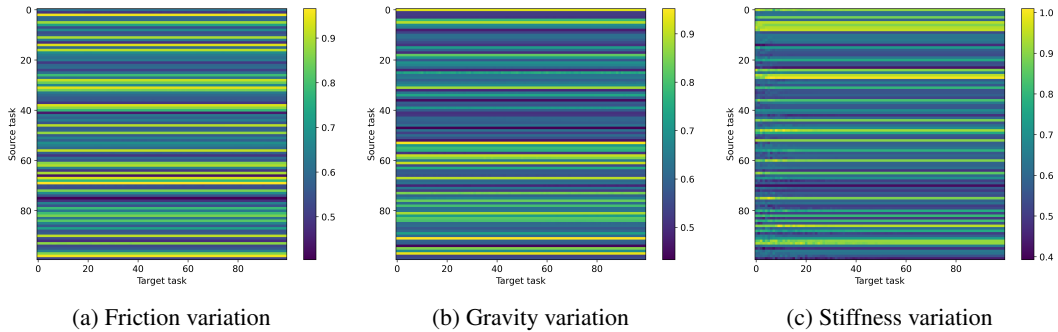
Figure 35: Examples of transferability heatmap for HalfCheetah.

Results Figure 36 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

32

environment. The trends across different parameters confirm the critical impact of task-specific dynamics on the effectiveness of the tested strategies.
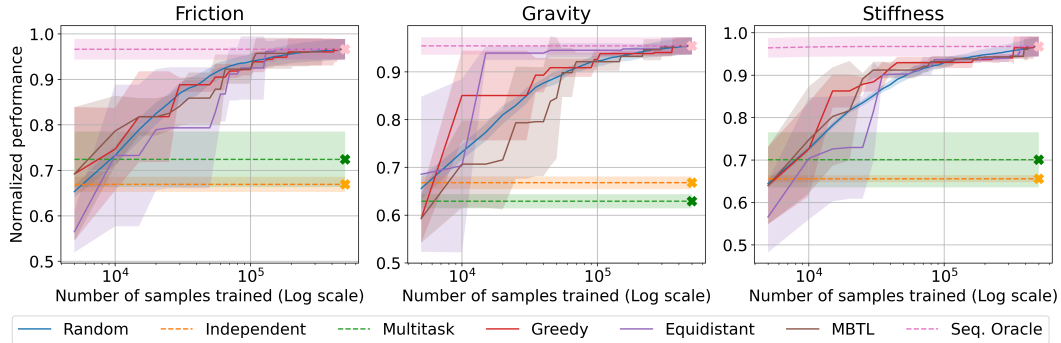
Figure 36: Comparison of normalized generalized performance of all target tasks: HalfCheetah.

A.4.13   Implementation of the recent multi-task baselines

In Figure 37, we compare two recent multi-task algorithms—PaCo [43] and MOORE [16]—with several our baselines tested on Cartpole CMDP benchmark. Although MOORE underperforms relative to our baseline multitask implementation, PaCo achieves competitive or even higher performance at certain points, demonstrating its potential to generalize across multiple tasks. Also, it is important to note that those algorithms are naively implemented without thorough investigation. These trends show that enhanced multi-task strategies can be beneficial in some CMDP settings, whereas not all multi-task methods readily adapt to broader parameter variations.
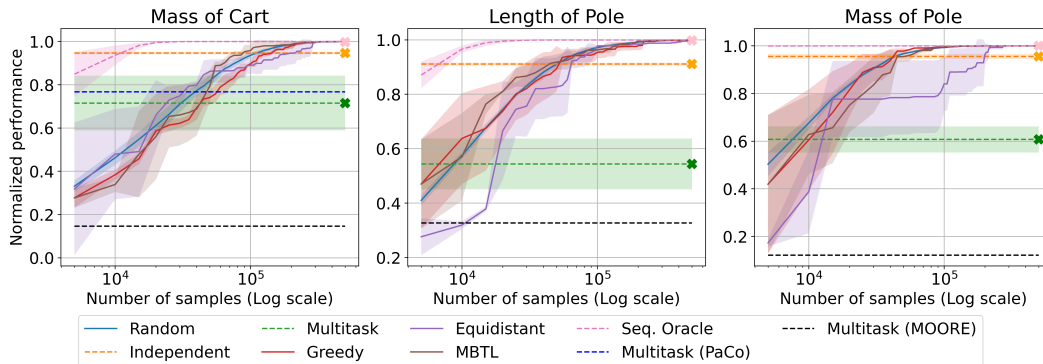
Figure 37: Normalized performance comparison of PaCo [43] and MOORE [16] on Cartpole benchmark.

33

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.

34

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction clearly state the contributions and scope, matching the theoretical and experimental results presented in the paper.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are discussed in Conclusion (Section 7).

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

35

Justification: The paper includes all necessary assumptions and provides complete proofs in the Appendix A.3.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: Detailed experimental settings, including models and hyperparameters, are provided to ensure reproducibility. Also, the code is submitted by the authors.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

36

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: The code are submitted by the authors with zip files, with detailed instructions for reproducing the results included in the supplemental material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyper-parameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The paper provides comprehensive details about the experiments, including data generation, hyperparameters, and optimizer types, to ease understanding of the results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Error bars and confidence intervals are reported for all significant experiments, with methods and assumptions clearly stated.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

37

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

Answer: [Yes]

Justification: The type of compute resources, memory, and execution time for each experiment are specified to ensure reproducibility in Appendix A.4.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
- The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
- The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

Answer: [Yes]

Justification: The research adheres to the NeurIPS Code of Ethics, with no deviations from the guidelines.

Guidelines:

- The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

Answer: [Yes]

Justification: The paper includes a section discussing both the potential positive and negative societal impacts of the research.

Guidelines:

38

- The answer NA means that there is no societal impact of the work performed.
- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper does not involve data or models with high risk for misuse.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: The paper uses the CARL framework under the Apache License 2.0, with proper credit given and license terms mentioned in the Appendix A.4.8.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.

39

- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The new assets, including code for the MBTL algorithm and experiments, are well-documented and submitted with an anonymized zip file.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing or research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

40

- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

41