Proceedings of Machine Learning Research vol vvv:1–27, 2025

Coreset-Based Task Selection for Sample-Efficient Meta-Reinforcement Learning

Donglin Zhan DZ2478@COLUMBIA.EDU

Leonardo F. Toso LT2879@COLUMBIA.EDU

James Anderson JAMES.ANDERSON@COLUMBIA.EDU

All authors are with the Department of Electrical Engineering, Columbia University, New York

Abstract

We study task selection for model-agnostic meta-reinforcement learning (MAML-RL) to enhance sample efficiency. Traditional meta-RL typically assumes that all available tasks are equally important, which can lead to task redundancy when they share similarities. To address this, we propose a coreset-based task selection approach that selects a weighted subset of tasks based on how diverse they are in the gradient space, prioritizing the most informative and diverse tasks. Such task selection reduces the number of samples needed to find an ϵ-close stationary solution by a factor of $\mathcal{O}(1/\epsilon)$. Consequently, it guarantees a faster adaptation to unseen tasks while focusing training on the most relevant tasks. As a case study, we incorporate task selection to MAML-LQR (Toso et al., 2024b), and prove a sample complexity reduction proportional to $\mathcal{O}(\log(1/\epsilon))$ when the task-specific reward additionally satisfy gradient dominance. Our theoretical guarantees underscore task selection as a key component for scalable and sample-efficient meta-RL.

Keywords: Meta-Reinforcement Learning; Task Selection; Model-free Learning; LQR

1. Introduction

Meta-reinforcement learning (meta-RL) has emerged as a powerful framework for learning policies that can quickly adapt to unseen environments (Wang et al., 2016; Finn et al., 2017). In particular, the model-agnostic meta-reinforcement learning (MAML-RL) algorithm has demonstrated success in enabling agents to learn a shared policy initialization that is only a few policy gradient steps away from optimality for any seen *and* unseen task (Duan et al., 2016; Nagabandi et al., 2018). Such quick adaptation is crucial, for example, in robotics (Song et al., 2020), where agents often need to operate in dynamic environments and provide and accomplish a variety of goals.

MAML-RL and meta-reinforcement learning more generally, typically assumes that all training tasks are equally important. This assumption may lead to task redundancy and excessive sampling costs as it is likely not worth sampling from multiple similar tasks; instead collecting data from a single representative task would suffice.

"Task selection" can be thought of a pre-processing step on the meta-learning pipeline. It seeks to identify a representative subset of tasks that captures the diversity across all training tasks, and then uses this smaller "coreset" for training. In particular, "coreset learning" has been proposed for task efficient training of machine learning models (Mirzasoleiman et al., 2020; Pooladzandi et al., 2022; Yang et al., 2023). Related work has also employed coreset selection to select clients in federated learning (Balakrishnan et al., 2022) and continual learning (Tiwari et al., 2022; Wang et al., 2022). For meta-learning, and in the context of classification tasks, Zhan and Anderson (2024) propose a data-efficient and robust task selection algorithm (DERTS) that outperforms existing sampling-based techniques. In essence, DERTS frames coreset learning as a submodular

This manuscript is a short version of our technical report. For any omitted details, please refer to its appendix.

optimization problem, where the goal is to select a subset of tasks that minimizes the maximum normed difference across task-specific gradients.

In the work above, it is assumed that task-specific gradients can be directly computed. In meta-RL, such an assumption may be restrictive as the meta-gradient depends on unknown task and trajectory distributions, where the later is also conditioned on the current policy. As such, it is challenging to compute the gradient via automatic differentiation (Rothfuss et al., 2018; Liu et al., 2019). To circumvent this, one must resort to gradient approximation, that by itself introduces extra difficulties to the analysis of meta-RL with task selection. Specifically, errors arising from gradient estimation and the meta-training on the coreset need to be carefully accounted for.

Contributions: Towards addressing these points, we propose a coreset-based task selection algorithm (inspired by `DERTS`) for meta-RL. The main contributions of our approach are:

• **Algorithmic:** This is the first work to propose a derivative-free coreset-based task selection approach for MAML-RL (Algorithm 1), which also comes with strong convergence guarantees (Section 3). We derive an ergodic convergence rate for non-concave task-specific reward functions (Theorem 1) and prove that Algorithm 1 finds an ϵ-close stationary solution after $N = \mathcal{O}(1/\epsilon)$ iterations when the task selection bias is made sufficiently small. We also incorporate task selection to meta-learning for control via the `MAML-LQR` algorithm (Toso et al., 2023b) and show that it learns a provably fast-to-adapt LQR controllers (Theorem 2) within $N = \mathcal{O}(\log(1/\epsilon))$ iterations while reducing task redundancy.

• **Sample Complexity:** We demonstrate that selecting a weighted subset of the most informative tasks reduces the sample complexity for achieving local convergence by a factor of $\mathcal{O}(1/\epsilon)$ (Corollary 1). In particular, this reduction is guaranteed when the set of training tasks is sufficiently large and tasks therein are sufficiently similar. Moreover, Algorithm 1 offers a sample complexity reduction proportional to $\mathcal{O}(\log(1/\epsilon))$ in the `MAML-LQR` setting (Corollary 2).

Related Work: Meta-reinforcement learning has been extensively studied across several applications, including robot manipulation (Yu et al., 2017), locomotion (Song et al., 2020), and building energy control (Luna Gutierrez and Leonetti, 2020). Most relevant to our work is Song et al. (2019, 2020), which is derivative-free but treats all tasks equally, leading to task redundancy. Task weighting is addressed in Zhan and Anderson (2024); Shin et al. (2023) by selecting representative task subsets. However, Zhan and Anderson (2024) focuses on classification tasks and simplifies gradient approximation by using the model pre-activation outputs, while (Shin et al., 2023) employs an information-theoretic metric for task selection and does not consider gradient-based training.

Finally, in the context of control, the linear quadratic regulator (LQR) problem has become a key baseline for policy optimization in reinforcement learning. In particular, (Molybog and Lavaei, 2021; Musavi and Dullerud, 2023; Toso et al., 2024b; Aravind et al., 2024; Pan and Zhu, 2024) study the meta-LQR problem and provide conditions for provably learning fast-to-adapt LQR controllers. Building on this, our work integrates task selection into the MAML-LQR setting, demonstrating its effectiveness in reducing the sample complexity. A more detailed overview of related work is included in Section 6.1 of the appendix.

2. Preliminaries

We now introduce the MAML-RL problem (Finn et al., 2017) and formalize our coreset-based task selection that selects a weighted subset of tasks based on their diversity in the gradient space.

2.1. Model Agnostic Meta-Reinforcement Learning

Let T_j be a reinforcement learning task drawn from a task distribution $\mathcal{P}(\mathcal{T})$ over a set of tasks $\mathcal{T}$. Let $\mathcal{M}$ denote a set of M tasks drawn from $\mathcal{P}(\mathcal{T})$, i.e., $\mathcal{M} := \{T_j \mid j = 1, 2, \ldots, M\}$. The objective of MAML-RL is to learn a meta-policy $\pi_{\theta^\star}$ with meta-parameter $\theta^\star \in \Theta$, trained on $\mathcal{M}$, such that within a few policy gradient steps, $\pi_{\theta^\star}$ can be adapted to an unseen task-specific policy. In particular, we set tasks T_j with the same state and action spaces $\mathcal{S}, \mathcal{A}$. In addition, each task T_j is associated with a reward function R_j and a Markov Decision Process (MDP) with $q_j(s_{t+1}|s_t, a_t)$ as the transition distribution at time step t. Also, let $J(\theta)$ and $J_j(\theta)$ be the MAML and task-specific reward functions, respectively. The one-shot MAML-RL problem is written as follows:

$$\theta^\star := \text{argmax}_{\theta\in\Theta}\, J(\theta) := \mathbb{E}_{T_j\sim\mathcal{P}(\mathcal{T})} J_j(\theta + \eta_{\text{inn}}\nabla J_j(\theta)), \tag{1}$$

for some positive inner step-size η_{inn}. Moreover, $J_j(\theta) := \mathbb{E}_{\bar{\tau}\sim\mathcal{P}_{T_j}(\bar{\tau}|\bar{\theta})}\mathcal{R}_j(\bar{\tau})$ is the task-specific reward incurred by π_θ, where $\bar{\theta} := \theta + \eta_{\text{inn}}\nabla_\theta\mathbb{E}_{\tau\sim\mathcal{P}_{T_j}(\tau|\theta)}[R_j(\tau)]$, and $\mathcal{P}_{T_j}(\tau|\theta)$ is the distribution of trajectories τ conditioned on the policy π_θ. Hence, the gradient-based MAML-RL updates follows: $\theta \leftarrow \theta + \eta_{\text{out}}\nabla J(\theta)$ with η_{out} denoting the outer step-size, and the MAML gradient given by

$$\nabla J(\theta) = \mathbb{E}_{T_j\sim\mathcal{P}(\mathcal{T})}\left[\mathbb{E}_{\bar{\tau}\sim\mathcal{P}_{T_j}(\bar{\tau}|\bar{\theta})}(\nabla_{\bar{\theta}}\log\mathcal{P}_{T_j}(\bar{\tau}|\bar{\theta})R_j(\bar{\tau})\nabla\bar{\theta})\right], \tag{2}$$

$$\nabla_{\bar{\theta}} = I + \eta_{\text{out}}\left(\int \mathcal{P}_{T_j}(\tau|\theta)\nabla^2\log\pi_\theta(\tau)R_j(\tau) + \mathcal{P}_{T_j}(\tau|\theta)\nabla_\theta\log\pi_\theta(\tau)\nabla_\theta\log\pi_\theta(\tau)^\top R_j(\tau)d\tau\right).$$

The direct computation of $\nabla J(\theta)$ may be intractable due to the expectation over unknown task and trajectory distributions. Work by Liu et al. (2019); Rothfuss et al. (2018), and Song et al. (2019) has highlighted such challenging gradient computation and considered the setting where we draw multiple tasks T_j from task distribution $\mathcal{P}(\mathcal{T})$ and sample multiple trajectories τ_l by playing with π_θ to compute an empirical task-specific reward function $\hat{J}_j(\theta)$. Following Song et al. (2019, 2020), we propose a derivative-free method for estimating the task-specific and MAML gradients through querying/estimating task-specific rewards.

2.2. Task Selection

Motivated by Yang et al. (2023); Zhan and Anderson (2024), we argue that not all the tasks in the task pool $\mathcal{M}$ are equally important for meta-training. Since multiple tasks may share similarities, that may lead to task redundancy and sample inefficiency, as it requires collecting data from multiple similar tasks when. In principle, collecting data from a single task that is representative of all the similar tasks should suffice for training. Our goal is to select a subset of tasks, $\mathcal{S}$ (the coreset), from the task pool $\mathcal{M}$ that best represents the diversity of the tasks in $\mathcal{M}$, such that the performance of the model trained on a weighted subset of $\mathcal{S}$ is sufficiently close to that of a model trained on the full task pool $\mathcal{M}$. In particular, we will prove that by carefully controlling the task selection bias one may achieve a substantial sample complexity reduction for both local and global convergence while also reducing task redundancy in meta-training.

The main steps of our meta training algorithm are: (i) coreset selection—selecting the coreset $\mathcal{S} \subset \mathcal{M}$, (ii) weight allocation for each task in $\mathcal{S}$ such that it captures the relative importance of the task, and (iii) meta-training on the coreset $\mathcal{S}$ and its corresponding weighting. In the following subsections we introduce the concept of gradient approximation over the task pools and establish the selection criterion for assigning tasks in $\mathcal{M}$ to $\mathcal{S}$.

2.2.1. FULL GRADIENT APPROXIMATION OVER THE TASK POOL

We aim to select the coreset $\mathcal{S} \subseteq \mathcal{M}$ with $\mathcal{S} := \{T_i \mid i = \alpha_1, \alpha_2, \ldots, \alpha_L\}$, where $\alpha_i \in [M]$ and $L \leq M$, with corresponding weights $\{\gamma_i \mid i = 1, 2, \ldots, L\}$, such that the gradient for training on $\mathcal{S}$ with corresponding weights γ_i approximates the meta-gradient on $\mathcal{M}$.

To better understand the coreset-based task selection we let $\Gamma : \mathcal{M} \to \mathcal{S}$ be a mapping from the task pool $\mathcal{M}$ to the coreset $\mathcal{S}$, i.e., that maps a task T_j from $\mathcal{M}$ to a task T_i in $\mathcal{S}$. For simplicity, we denote $\Gamma(T_j) = T_i$ as $\Gamma(j) = i$. In addition, let $\mathcal{S}_c$ denote the complement of $\mathcal{S}$ in $\mathcal{M}$. Following Mirzasoleiman et al. (2020), we define the weight γ_i of the selected task $T_i \in \mathcal{S}$ as $\gamma_i := \sum_{j\in\mathcal{M}} \mathbb{1}_{\{\Gamma(j)=i\}}$, where $\mathbb{1}_{\mathcal{C}}$ is the indicator function over some set $\mathcal{C}$. Also, the summation over $\mathcal{M}$ computes the number of tasks in the task pool that are assigned to task T_i in $\mathcal{S}$.

Then, by using the definition of the mapping function Γ the gradient approximation error due to training over $\mathcal{S}$ instead of $\mathcal{M}$ is given by

$$\left\|\sum_{j\in\mathcal{M}} \nabla J_j(\theta) - \sum_{j\in\mathcal{M}} \nabla J_{\Gamma(j)}(\theta)\right\| = \left\|\sum_{j\in\mathcal{M}} \nabla J_j(\theta) - \sum_{i\in\mathcal{S}} \gamma_i \nabla J_i(\theta)\right\| \leq \sum_{j\in\mathcal{M}} \min_{i\in\mathcal{S}} \|\nabla J_j(\theta) - \nabla J_i(\theta)\|, \tag{2}$$

where our objective is to control and make this error as small as possible (by selecting $\mathcal{S}$ and the weights). We emphasize that we *do not* have access to $\mathcal{S}$ and subsequently Γ. Namely, we cannot directly evaluate that error, and optimizing over the subset of tasks $\mathcal{S}$ is NP-hard. Instead, we proceed by developing an algorithm that seeks to minimize the RHS of (2). Assuming the elements in $\mathcal{S}$ are fixed, we assign each task in $\mathcal{M}$ to its closest element in $\mathcal{S}$, in the gradient space, through the mapping Γ. To do so, the weights γ_i, for all tasks $T_i \in \mathcal{S}$, corresponding to the mapping Γ can be allocated as $\gamma_i = \sum_{j\in\mathcal{M}} \mathbb{1}_{\left\{j=\text{argmin}_{T_i\in\mathcal{S}}\|\nabla J_j(\theta)-\nabla J_i(\theta)\|\right\}}$.

However, as previously discussed, directly computing $\nabla J_i(\theta)$ is not tractable for most RL tasks. This is due to the fact that simulating trajectories and performing backpropagation through deep-RL models incurs a large computational cost. Motivated by this, to estimate the task-specific gradients we resort to a derivative-free task selection approach based on a zeroth-order optimization scheme. In particular, we consider a two-point estimation since it has a lower estimation variance compared to its one-point counterpart (Malik et al., 2019). Zeroth-order estimation is a Gaussian smoothing approach (Nesterov and Spokoiny, 2017) based on Stein's identity (Stein, 1972) that relates gradient to reward queries. We refer the reader to Flaxman et al. (2004); Spall (2005) for further details on zeroth-order estimation. The two-point zeroth-order estimation of $\nabla J_i(\theta)$ is

$$\texttt{ZO2P}(\theta, n_s, r) = \widehat{\nabla} J_i(\theta) := \frac{d}{2n_s r^2} \sum_{l=1}^{n_s} \left(J_i(\theta + v_l) - J_i(\theta - v_l)\right) v_l, \tag{3}$$

with $r > 0$ denoting the smoothing radius, $v_l \in \mathbb{R}^d$ randomly drawn from a uniform distribution over the Euclidean sphere of radius r, $\mathbb{S}_r^{d-1}$, namely, $v_l \sim \mathbb{S}_r^{d-1}$ and n_s the number of samples. In addition, $d := d_1 \times d_2$ is for LQR case. Finally, we define the estimation of the meta-gradient over $\mathcal{M}$ and $\mathcal{S}$ as follows:

$$\nabla_{\mathcal{M}} J(\theta) := \frac{1}{M} \sum_{j\in\mathcal{M}} g_j(\theta), \quad \nabla_{\mathcal{S}} J(\theta) := \frac{1}{M} \sum_{i\in\mathcal{S}} \gamma_i g_i(\theta),$$

where $g_i(\theta) := \frac{d}{2r^2 n_s} \sum_{l=1}^{n_s} \Big(J_i(\theta + u_l + \eta_{\text{inn}} \widehat{\nabla} J_i(\theta)) - J_i(\theta - u_l + \eta_{\text{inn}} \widehat{\nabla} J_i(\theta))\Big) u_l$.

2.2.2. CORESET-BASED TASK SELECTION

We note that minimizing the RHS of (2) is mathematically equivalent to maximizing the facility location function, which is a well-known submodular function (Cornuejols et al., 1977).

Definition 1 (Submodularity Nemhauser et al. (1978)) *A set function $F : 2^V \to \mathbb{R}^+$ is submodular if $F(e \mid S) := F(S \cup \{e\}) - F(S) \geq F(T \cup \{e\}) - F(T)$, for any $S \subseteq T \subseteq V$ and $e \in V \backslash T$. F is monotone if $F(e \mid S) \geq 0$ for any $e \in V \backslash \bar{S}$ and $S \subseteq V$.*

We leverage Definition 1 and (2) to define a monotone submodular function $\mathcal{F}$ over $\mathcal{S}$ with respect to the zeroth-order approximated gradient $g_i(\theta)$. That is,

$$\mathcal{F}(\mathcal{S}) := C - \sum_{j \in \mathcal{M}} \min_{i \in \mathcal{S}} \|g_j(\theta) - g_i(\theta)\|, \tag{4}$$

where $C > 0$ upper bounds $\mathcal{F}(\mathcal{S})$. Therefore, to formulate our task selection objective, we restrict the cardinality of $\mathcal{S}$ and make the number of tasks in $\mathcal{S}$ sufficiently small (as preferred $L \ll M$), i.e., this is introduced as the constraint $|\mathcal{S}| \leq L$ in the following submodular optimization. Then, $\mathcal{S}$ is learned by solving

$$\mathcal{S}^\star = \arg\max_{\mathcal{S} \subseteq \mathcal{M}} \mathcal{F}(\mathcal{S}), \text{ s.t. } |\mathcal{S}| \leq L. \tag{5}$$

It is well-known that (5) can be solved through a greedy-based approach with a $1 - e^{-1}$ error bound on the corresponding approximate solution (Nemhauser et al., 1978; Wolsey, 1982). We then incorporate such coreset-based task selection in the MAML-RL training over the learned weighted subset $\mathcal{S}$ in Algorithm 1. To start with, we initialize $\mathcal{S}$ as the empty set in step 1, and for each greedy iteration, we select a task T_i from $\mathcal{S}_c$ that maximizes the marginal utility $\mathcal{F}(T_i|\mathcal{S}) = \mathcal{F}(\mathcal{S} \cup T_i) - \mathcal{F}(\mathcal{S})$ in steps 7 and 8, where the update step for $\mathcal{S}$ can be described as $\mathcal{S} = \mathcal{S} \cup \arg\max_{T_i \in \mathcal{S}_c} \mathcal{F}(T_i \mid \mathcal{S})$. With the learned subset $\mathcal{S}$ in hand, the weights γ_i for all tasks $T_i \in \mathcal{S}$ are computed in step 10. The task selection is then followed by step 12 to 17 where MAML is applied on the coreset $\mathcal{S}$. In the next section we present the theoretical guarantees of Algorithm 1.

3. Theoretical Guarantees

We first introduce the ergodic convergence rate (i.e., local convergence analysis) of Algorithm 1 for the general case of non-concave task-specific reward function. We then extend our results to meta-learning for control, specifically by applying task selection to the `MAML-LQR` algorithm Toso et al. (2024b), and derive global convergence guarantees when the task-specific reward satisfies a gradient dominance property. In addition, we discuss the sample reduction benefit of our coreset-based task selection for both MAML-RL and MAML-LQR problems.

3.1. Ergodic Convergence Rate

For the local convergence analysis, i.e., when $J_i(\theta)$ is generally non-concave, our goal is to characterize the ergodic convergence rate of Algorithm 1 with respect to the MAML reward function (1), namely, to control $\frac{1}{N} \sum_{n=0}^{N-1} \|\nabla J(\theta_n)\|^2$. We first assume that the task-specific reward function and its gradient are locally smooth and that the gradient is uniformly upper bounded. In addition, we assume that the upper bound of $\mathcal{F}(\mathcal{S})$ can be made sufficiently small.

Assumption 1 *(Local smoothness) The task-specific reward function $J_i(\theta)$ and its gradient $\nabla J_i(\theta)$ are smooth with constants β and ψ, respectively, i.e., for for any $\theta, \theta' \in \Theta$, we have*

$$|J_i(\theta) - J_i(\theta')| \leq \beta J_i(\theta) \|\theta - \theta'\|, \quad \|\nabla J_i(\theta) - \nabla J_i(\theta')\| \leq \psi \|\theta - \theta'\|. \tag{6}$$

Algorithm 1 `Coreset Selection for MAML-RL`

Input: initial meta-policy parameter θ_0; step-sizes $\eta_{\text{inn}}, \eta_{\text{out}}$; number of samples n_s; smoothing radius r; number of iterations N; number of selected tasks L; task pool $\mathcal{M}$; coreset $\mathcal{S} = \emptyset$
Coreset Selection:
for all tasks T_j in $\mathcal{M}$ **do**
  $\widehat{\nabla} J_j(\theta_0) \leftarrow \texttt{ZO2P}(\theta_0, n_s, r),\ g_j(\theta_0) \leftarrow \texttt{ZO2P}(\theta_0 + \eta_{\text{inn}} \widehat{\nabla} J_j(\theta_0), n_s, r)$ ▷ `estimation`
end for
while $|\mathcal{S}| < L$ **do**
  $T_i \in \operatorname{argmax}_{T_i \in \mathcal{S}^C} \mathcal{F}(T_i \mid \mathcal{S})$
  $\mathcal{S} = \mathcal{S} \cup \{T_i\}$
end while
$\gamma_i = \sum_{j \in \mathcal{M}} \mathbb{1}_{\{j = \operatorname{argmin}_{T_i \in \mathcal{S}} \|g_j(\theta_0) - g_i(\theta_0)\|\}}$ ▷ `weight allocation`
MAML-RL over $\mathcal{S}$:
for all iteration $n = \{0, 1, \ldots, N-1\}$ **do**
  for all tasks T_i in $\mathcal{S}$ **do**
    $\widehat{\nabla} J_i(\theta_n) \leftarrow \texttt{ZO2P}(\theta_n, n_s, r),\ \ g_i \leftarrow \texttt{ZO2P}(\theta_n + \eta_{\text{inn}} \widehat{\nabla} J_i(\theta_n), n_s, r)$ ▷ `estimation`
  end for
  $\nabla_{\mathcal{S}} J(\theta_n) = \frac{1}{M} \sum_{i \in \mathcal{S}} \gamma_i g_i(\theta_n),\ \ \theta_{n+1} = \theta_n + \eta_{\text{out}} \nabla_{\mathcal{S}} J(\theta_n)$ ▷ `meta update`
end for
Output: θ_N

Assumption 2 *(Gradient uniform bound) $\|\nabla J_i(\theta)\| \leq \phi$, for any $T_i \in \mathcal{P}(\mathcal{T})$ and $\theta \in \Theta$.*

Assumptions 1 and 2 are standard in the convergence analysis of training dynamics (Oymak and Soltanolkotabi, 2019; Liu et al., 2022), as well as in the literature of stochastic gradient descent (SGD) for non-convex loss functions (Stich, 2018; Li et al., 2019). Later, for the MAML-LQR setting, such conditions are in fact properties of the task-specific LQR cost.

Assumption 3 *The constant C in (4) is set sufficiently small, i.e., $C = \mathcal{O}(\epsilon)$, for some small ϵ.*

It is also worth noting that making C sufficiently small is standard in coreset learning for data-efficient machine-learning (Mirzasoleiman et al., 2020; Yang et al., 2023) as it guarantees that task selection estimation error remains sufficiently small. Next, we define the maximum normed difference between the gradient of two task-specific reward functions over the parameter space and present the local convergence guarantee of Algorithm 1.

Definition 2 *The maximum normed difference between two distinct task-specific gradients over the parameter space $\theta \in \Theta$ is $\xi_{i,j} := \max_{\theta \in \Theta} \|\nabla J_i(\theta) - \nabla J_j(\theta)\|$.*

Theorem 1 *(Stationary solution) Suppose Assumptions 1, 2 and 3 are satisfied. In addition, suppose the number of samples and smoothing radius are set according to*

$$n_s \geq C_{approx,1} \left(\frac{\sigma^2}{\epsilon^2} + \frac{b}{3\epsilon} \right) \log \left(\frac{d_1 + d_2}{\delta} \right), \quad r \leq \frac{\epsilon}{C_{approx,1} \psi},$$

with $\sigma^2 := (d\beta J_{\max})^2 + (\epsilon + \phi)^2$ and $b := d\beta J_{\max} + \epsilon + \phi$ for $J_{\max} := \max_{T_j \in \mathcal{M}, \theta \in \Theta} J_j(\theta)$, $\delta, \epsilon \in (0, 1)$, and a sufficiently large universal constant $C_{approx,1}$. Lastly, suppose that the step-sizes *scale as* $\eta_{inn} = \mathcal{O}(\epsilon/d)$ *and* $\eta_{out} = \mathcal{O}(1)$ *and* $L = \mathcal{O}(1)$*. Then, Algorithm 1 satisfies*

$$\frac{1}{N}\sum_{n=0}^{N-1}\|\nabla J(\theta_n)\|_2^2 \leq \mathcal{O}\left(\frac{\Delta_0}{\eta_{out} N} + \left(\frac{1}{M}\sum_{j\in\mathcal{M}}\min_{i\in\mathcal{S}^\star}\xi_{i,j}\right)^2\right),$$

with probability $1-\delta$*, and initial MAML-RL optimality gap* $\Delta_0 = J(\theta^\star) - J(\theta_0)$*.*

Task selection bias: We emphasize that the bias in the ergodic convergence rate comes from the task selection (steps 2-10 of Algorithm 1), and it can be made sufficiently small for $L = \mathcal{O}(1)$. Namely, there exists an L for which that bias is minimized given arbitrarily different tasks in $\mathcal{M}$. For instance, consider the worst-case scenario where the tasks in $\mathcal{M}$ are all substantially different from each other. Then, $L = M$ guarantees that such bias is zero while recovering the convergence rate for the setting without subset selection. Hence, for the case where there are sufficiently similar tasks in $\mathcal{M}$, we let the practitioner to set $L \ll M$ and ensure that such bias remains negligible.

Let $\mathcal{S}_c^{\mathcal{S}} := \frac{L}{M}\mathcal{S}_c^{\mathcal{M}} + \mathcal{O}(Mn_s)$ and $\mathcal{S}_c^{\mathcal{M}} := \mathcal{O}(MNn_s)$ denote the total number of samples in Algorithm 1 to find an ϵ-near stationary solution, with and without task selection, respectively.

Corollary 1 *(Sample complexity) Let the arguments of Theorem 1 hold. Suppose the number of iterations scales as* $N = \mathcal{O}(1/\epsilon)$ *and the number of tasks in the task pool is sufficiently large as* $M = \mathcal{O}(1/\epsilon)$*. Therefore, our coreset-based task selection offers a sample complexity reduction such as* $\mathcal{S}_c^{\mathcal{M}} = \mathcal{O}(1/\epsilon)\mathcal{S}_c^{\mathcal{S}}$*, with high probability.*

Task selection trade-off: It is also worth highlighting the trade-off of selecting L in a heterogeneous task regime. It is evident that by setting L small, when M is sufficiently large, will be beneficial for reducing the number of samples when $\mathcal{O}\left(\frac{1}{M}\sum_{j\in\mathcal{M}}\min_{i\in\mathcal{S}^\star}\xi_{i,j}\right)$ is sufficiently small (i.e., in the order of ϵ). However, when the tasks are sufficiently different, setting L small may even prevent convergence to a stationary solution. We refer the reader to (Mirzasoleiman et al., 2020; Yang et al., 2023) for algorithmic alternatives that do not require a pre-specified L.

Discussion: Theorem 1 and Corollary 1 summarize our main results for the MAML-RL setting. In particular, in Theorem 1, $\frac{1}{N}\sum_{n=0}^{N-1}\|\nabla J(\theta_n)\|_2^2$ is controlled by two terms. The first term scales as $\mathcal{O}\left(\frac{\Delta_0}{\eta_{\text{out}} N}\right)$ and it refers to the complexity of finding a stationary solution given the initial meta-policy parameters θ_0. On the other hand, as previously discussed, $\mathcal{O}\left(\frac{1}{M}\sum_{j\in\mathcal{M}}\min_{i\in\mathcal{S}^\star}\xi_{i,j}\right)$ is due to the meta-training over the weighted subset of tasks $\mathcal{S}$ instead of the entire task pool $\mathcal{M}$.

Although Mirzasoleiman et al. (2020); Yang et al. (2023) also highlight the effect of the additive bias in the context of coresets for data-efficient deep-learning, they assume the direct computation of gradients which simplifies the setting and prevents characterization of the sample complexity and subsequently the benefit of task selection. We fill that gap for the MAML-RL problem and stress that Corollary 1 is not an artifact of the zeroth-order gradient estimation scheme used in this work, and it may be extended to any derivative-free approaches (Salimans et al., 2017).

Proof idea: The main step in the proof of Theorem 1 is to control the gradient estimation error $\|\nabla J(\theta) - \nabla_{\mathcal{S}} J(\theta)\|$ for any $\theta \in \Theta$. To do so, we first observe that

$$\|\nabla J(\theta) - \nabla_{\mathcal{S}} J(\theta)\| \leq \underbrace{\|\nabla J(\theta) - \nabla_{\mathcal{M}} J(\theta)\|}_{\text{Zeroth-order estimation error}} + \underbrace{\|\nabla J_{\mathcal{M}}(\theta) - \nabla_{\mathcal{S}} J(\theta)\|}_{\text{Task selection bias}},$$

where the zeroth-order estimation error can be controlled by making n_s sufficiently large and r sufficiently small through matrix concentration inequalities (Tropp, 2012). Moreover, we control the task selection bias by first using the fact that $\mathcal{F}(S) \geq (1-e^{-1})\mathcal{F}(\mathcal{S}^\star)$ (Nemhauser et al., 1978). Then, by also controlling the estimation error in $g_j(\theta_0)$ and $g_i(\theta_0)$ and making C sufficiently small, $\|\nabla J_{\mathcal{M}}(\theta) - \nabla_{\mathcal{S}} J(\theta)\| \lesssim \epsilon + \frac{1}{M}\sum_{j\in\mathcal{M}} \min_{i\in\mathcal{S}^\star} \xi_{i,j}$, which can also be made sufficiently small by carefully tuning L. Subsequent proof steps follows from Assumptions 1 and 2. We refer the reader to Appendix 6.3, 6.4 and 6.5 for the detailed proof. Next, we consider the MAML-LQR problem and discuss the benefit of task selection in the setting where $J_j(\theta)$ satisfies gradient dominance.

3.2. Linear Quadratic Regulator (LQR) Problem

Consider the MAML-LQR problem from Toso et al. (2024b), i.e., the task pool $\mathcal{M}$ is composed of M distinct LQR tasks $T_j = (A_j, B_j, Q_j, R_j)$ with systems matrices $A_j \in \mathbb{R}^{d_1\times d_1}$, $B_j \in \mathbb{R}^{d_1\times d_2}$, and cost matrices $Q_j \in \mathbb{S}^{d_1}_{\succ 0}$, $R_j \in \mathbb{S}^{d_2}_{\succ 0}$, for any $j \in [M]$. In particular, each task $T_j \in \mathcal{M}$ is equipped with the objective of designing a controller $K_j^\star$ that solves

$$K_j^\star = \underset{K\in\mathcal{K}_j}{\operatorname{argmin}} \left\{ J_j(K) := \mathbb{E}\left[\sum_{t=0}^{\infty} x_t^{(j)\top}\left(Q_j + K^\top R_j K\right) x_t^{(j)}\right]\right\} \text{ s.t. } x_{t+1}^{(j)} = (A_j - B_j K) x_t^{(j)}, \quad (7)$$

where $\mathcal{K}_j := \{K \mid \rho(A_j - B_j K) < 1\}$ denotes the task-specific stabilizing set of controllers.

The objective of MAML-LQR is to design $K^\star$ that stabilizes any LQR task drawn from $\mathcal{P}(\mathcal{T})$, *and*, $K^\star$ should only be a few PG steps away from any unseen task-specific optimal controller. Similar to (1), with $\theta = K$, the MAML-LQR problem is:

$$\begin{aligned} K^\star = {} & \operatorname{argmax}_{K\in\overline{\mathcal{K}}} J(K) := \mathbb{E}_{T_j\sim\mathcal{P}(\mathcal{T})} J_j(K - \eta_{\text{inn}} \nabla J_j(K)) \\ \text{s.t.} \quad & x_{t+1}^{(j)} = (A_j - B_j K) x_t^{(j)}, \forall T_j \sim \mathcal{P}(\mathcal{T}), \end{aligned} \quad (8)$$

where $\overline{\mathcal{K}} := \cap_{T_j\sim\mathcal{P}(\mathcal{T})} \mathcal{K}_j$ denotes the MAML-LQR stabilizing set. We note that the crucial difference between (1) and (8) is the necessity for designing a controller $K \in \overline{\mathcal{K}}$ that *stabilizes* any system (A_j, B_j) drawn from the distribution of tasks $T_j \sim \mathcal{P}(\mathcal{T})$. Later, we show that Algorithm 1 produces such stabilizing controllers, while also reducing task redundancy with task selection.

We emphasize that our goal is to understand and characterize the benefit of task selection (Algorithm 1), for learning stabilizing controllers that can quickly adapt to unseen tasks in the LQR setting. For this purpose, and following Toso et al. (2024b), we next define the task specific and MAML-LQR stabilizing sub-level sets, as well as re-state the smoothness, gradient dominance and task heterogeneity properties of the LQR problem.

Definition 3 *(Stabilizing sub-level sets) For any task $T_j \sim \mathcal{P}(\mathcal{T})$, the task-specific sub-level set $\mathcal{G}_j^\mu \subseteq \mathcal{K}_j$ is $\mathcal{G}_j^\mu := \left\{K \mid J_j(K) - J_j(K_j^\star) \leq \mu\Delta_0^{(j)}\right\}$, with $\Delta_0^{(j)} = J^{(j)}(K_0) - J^{(j)}(K_j^\star)$, $\mu > 0$. In addition, the MAML-LQR stabilizing sub-level set is $\mathcal{G} := \cap_{j\sim\mathcal{P}(\mathcal{T})} \mathcal{G}_j^\mu \subseteq \overline{\mathcal{K}}$.*

Assumption 4 *(Initial stabilizing controller) $K_0 \in \mathcal{G}$* [1].

Assumption 5 *(Task heterogeneity) For any two distinct tasks $T_i, T_j \sim \mathcal{P}(\mathcal{T})$ we have that*

$$\max_{i\neq j}\|A_i - A_j\| \leq \epsilon_A, \max_{i\neq j}\|B_i - B_j\| \leq \epsilon_B, \max_{i\neq j}\|Q_i - Q_j\| \leq \epsilon_Q, \max_{i\neq j}\|R_i - R_j\| \leq \epsilon_R,$$

where$\epsilon_A, \epsilon_B, \epsilon_Q, \epsilon_R \geq 0$. We further denote $\epsilon_{het} = (\epsilon_A, \epsilon_B, \epsilon_Q, \epsilon_R)$.

1. As stressed in Toso et al. (2024b) `MAML-LQR` must be initialized from an stabilizing controller to produce finite costs and subsequently well-defined gradient estimations.

Lemma 4 (Lemma 4 from Toso et al. (2024b)) *For any two distinct tasks $T_i, T_j \sim \mathcal{P}(\mathcal{T})$ and stabilizing controller $K \in \mathcal{G}$. It holds that, $\|\nabla J_i(K) - \nabla J_j(K)\| \leq f(\epsilon_{het})$, where $f(\epsilon_{het})$ denotes the gradient heterogeneity bias.*

Lemma 5 *Given any task $T_j \sim \mathcal{P}(\mathcal{T})$ and stabilizing controllers $K, K' \in \mathcal{G}$ such that $\|\Delta\| := \|K' - K\|_F < \infty$. It holds that $\|\nabla J_j(K)\|_F \leq \phi$,*

$$\left|J_j\left(K'\right) - J_j(K)\right| \leq \beta J_j(K)\|\Delta\|_F, \quad \left\|\nabla J_j\left(K'\right) - \nabla J_j(K)\right\|_F \leq \psi\|\Delta\|_F,$$

and $\|\nabla J_j(K)\|_F^2 \geq \lambda_j(J_j(K) - J_j(K_j^\star))$, where $\lambda_j > 0$ denotes the gradient dominance constant.

We remark that Lemma 5 was initially proved in Fazel et al. (2018) and subsequently revisited in Gravell et al. (2020); Wang et al. (2023); Toso et al. (2024b), where the explicit expression of the problem dependent constants ϕ, β, ψ are provided.

Theorem 2 *(Gap to optimality) Suppose that Assumptions 3, 4 and 5 hold. In addition, suppose that the inner and outer step-sizes are of the order $\eta_{inn} = \mathcal{O}(\epsilon/d)$ and $\eta_{out} = \mathcal{O}(1)$, and that the number of samples and smoothing radius are set according to*

$$n_s \geq C_{approx,2} \min(d_1, d_2)\left(\frac{\sigma^2}{\epsilon^2} + \frac{b}{3\sqrt{\min(d_1, d_2)\epsilon}}\right)\log\left(\frac{d_1 + d_2}{\delta}\right), \quad r \leq \frac{\epsilon}{C_{approx,2}\psi},$$

where $C_{approx,2}$ is a sufficiently large universal constant. Then, when combined with task selection the MAML-LQR satisfies

$$J_j(K_N) - J_j(K^\star) \leq \left(1 - \frac{\lambda_j \eta_{out}}{4}\right)^N \Delta_0^{(j)} + \mathcal{O}(f^2(\epsilon_{het})),$$

with probability $1 - \delta$, for any task $T_j \sim \mathcal{P}(\mathcal{T})$ with $\Delta_0^{(j)} = J_j(\theta^\star) - J_j(\theta_0)$.

Let $\bar{\mathcal{S}}_c^{\mathcal{S}} = \frac{L}{M}\bar{\mathcal{S}}_c^{\mathcal{M}} + \mathcal{O}(Mn_s)$ and $\bar{\mathcal{S}}_c^{\mathcal{M}} = \mathcal{O}(MNn_s)$ denote the total number of samples required in Algorithm 1 to learn a LQR controller that is ϵ-close to any task-specific optimal controller up to a heterogeneity bias, with and without task selection, respectively.

Corollary 2 *(Sample complexity) Let the arguments of Theorem 2 hold with $L = \mathcal{O}(1)$. Suppose the number of iterations scales as $N = \mathcal{O}(\log(1/\epsilon))$ and the number of tasks in the task pool is sufficiently large as $M = \mathcal{O}(\log(1/\epsilon))$. Then, using task selection one may reduce the sample complexity to $\bar{\mathcal{S}}_c^{\mathcal{M}} = \mathcal{O}(\log(1/\epsilon))\bar{\mathcal{S}}_c^{\mathcal{S}}$, with high probability.*

The proofs are in Appendix 6.6 and 6.7. We note that task selection does not affect the ability of `MAML-LQR` trained on $\mathcal{S}$ to produce stabilizing controllers, i.e., $K_n \in \mathcal{G}$ for any iteration. We defer the stability analysis to Appendix 6.8. We remark that both Algorithm 1 and `MAML-LQR` (Toso et al., 2024b, Algorithm 3) converges to a controller that is ϵ-close to each task-optimal controller up to a heterogeneity bias. However, by selecting a weighted set of the most informative tasks $\mathcal{S}$, the sample complexity of learning such meta-controller can be reduced by a factor of $\mathcal{O}(\log(1/\epsilon))$.

4. Numerical Validation

Reinforcement Learning - Cart Pole[2]: We evaluate Algorithm 1 in a deep meta-RL setting. In particular, we examine the cart pole environment (Towers et al., 2024), a classical control environment where physical properties of the system vary across tasks, including cart mass, pole mass, and pole length. The policy is parameterized by a multi-layer perceptron (MLP) architecture consisting of two hidden layers. Further details are provided in Appendix 6.9. Figure 1 shows the learning curves comparing Algorithm 1 with vanilla MAML (Finn et al., 2017), depicting both average rewards and standard deviations across 5 runs.

2. Code can be downloaded from: `https://github.com/jd-anderson/Coreset-meta-RL`.

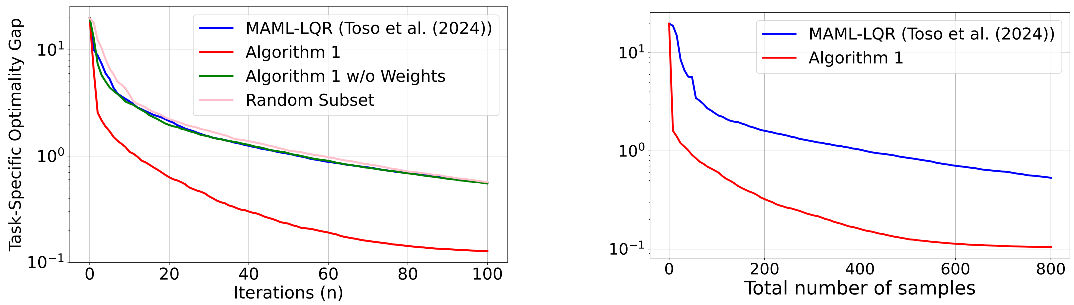

Figure 2: Optimality gap of Algorithm 1 in the MAML-LQR setting with respect to iterations.

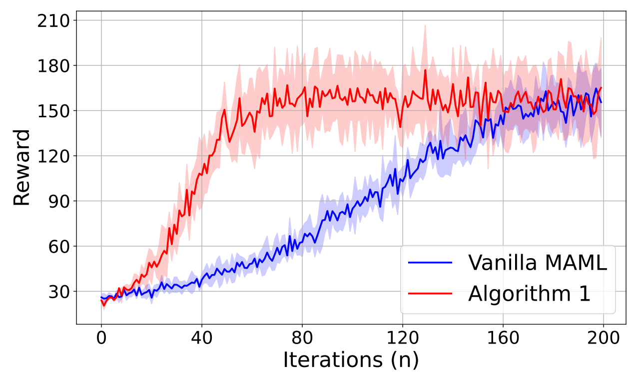

Figure 1: Reward comparison of Algorithm 1 and the vanilla MAML (Finn et al., 2017) on RL cart pole tasks.

Our results demonstrate that Algorithm 1 learns approximately $3\times$ faster than the vanilla MAML algorithm, reaching higher reward values in fewer iterations. While our approach exhibits slightly higher variance (as indicated by the larger shaded regions representing standard deviation), it consistently outperforms the baseline in terms of learning speed. Notably, the task selection method reaches a reward of 150 around iteration 60, whereas MAML takes approximately 200 iterations to research the same reward. These empirical findings strongly support our theoretical analysis regarding sample complexity reduction (Corollary 1), validating that careful task selection significantly enhance sample-efficiency in meta-RL.

Linear Quadratic Regulator: We follow the setting proposed by Toso et al. (2024b) to validate our theoretical guarantees in the MAML-LQR setting. In particular, Figure 2 (left) shows the optimality gap across iterations. We implemented the MAML-LQR on three scenarios: the full task pool (40 tasks), a selected subset (10 tasks), and two ablation baselines - selected subsets without weight assignments and randomly selected subsets. Our results demonstrate the faster convergence on the weighted selected subset, while both unweighted selected subset and random subset achieves at most the same performance as the full task pool. Moreover, Figure 2 (right) depicts the optimal cost gap between the selected subset and full task pool with respect to sample size, confirming our theoretical results on sample complexity reduction for the MAML-LQR (Corollary 2).

5. Conclusions and Future Work

We proposed a coreset-based task selection to enhance sample efficiency in meta-RL. By prioritizing the most informative and diverse tasks, Algorithm 1 addressed the task redundancy of traditional meta-RL. We demonstrated that task selection reduces the sample complexity of finding ϵ-near optimal solutions for both MAML-RL (i.e., by a factor of $\mathcal{O}(1/\epsilon)$) and MAML-LQR (i.e., proportional to $\mathcal{O}(1/\epsilon)$), which are further validated through deep-RL and LQR experiments. Future work may explore adding a clustering layer (Toso et al., 2023a) based on the task selection weights to the meta-RL pipeline in order to alleviate the heterogeneity bias arising in the MAML-LQR setting.

Acknowledgments

Leonardo F. Toso is funded by the Center for AI and Responsible Financial Innovation (CAIRFI) Fellowship and by the Columbia Presidential Fellowship. James Anderson is partially funded by NSF grants ECCS 2144634 and 2231350.

REFERENCES

Ashwin Aravind, Mohammad Taha Toghani, and César A Uribe. A moreau envelope approach for lqr meta-policy estimation. *arXiv preprint arXiv:2403.17364*, 2024.

Karol Arndt, Murtaza Hazara, Ali Ghadirzadeh, and Ville Kyrki. Meta reinforcement learning for sim-to-real domain adaptation. In *2020 IEEE international conference on robotics and automation (ICRA)*, pages 2725–2731. IEEE, 2020.

Ravikumar Balakrishnan, Tian Li, Tianyi Zhou, Nageen Himayat, Virginia Smith, and Jeff Bilmes. Diverse client selection for federated learning via submodular maximization. In *International Conference on Learning Representations*, 2022.

Gerard Cornuejols, Marshall Fisher, and George L Nemhauser. On the uncapacitated location problem. In *Annals of Discrete Mathematics*, volume 1, pages 163–177. Elsevier, 1977.

Yan Duan, John Schulman, Xi Chen, Peter L Bartlett, Ilya Sutskever, and Pieter Abbeel. RL^2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

Maryam Fazel, Rong Ge, Sham Kakade, and Mehran Mesbahi. Global convergence of policy gradient methods for the linear quadratic regulator. In *International conference on machine learning*, pages 1467–1476. PMLR, 2018.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pages 1126–1135. PMLR, 2017.

Abraham D Flaxman, Adam Tauman Kalai, and H Brendan McMahan. Online convex optimization in the bandit setting: gradient descent without a gradient. *arXiv preprint cs/0408007*, 2004.

Ali Ghadirzadeh, Xi Chen, Petra Poklukar, Chelsea Finn, Mårten Björkman, and Danica Kragic. Bayesian meta-learning for few-shot policy adaptation across robotic platforms. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1274–1280. IEEE, 2021.

Benjamin Gravell, Peyman Mohajerin Esfahani, and Tyler Summers. Learning optimal controllers for linear systems with multiplicative noise via policy gradient. *IEEE Transactions on Automatic Control*, 66(11):5283–5298, 2020.

Krishnateja Killamsetty, Ganesh Ramakrishnan, Abir De, and Rishabh Iyer. Grad-match: Gradient matching based data subset selection for efficient deep model training. In *International Conference on Machine Learning*, pages 5464–5474. PMLR, 2021.

Bruce Lee, Anders Rantzer, and Nikolai Matni. Nonasymptotic regret analysis of adaptive linear quadratic control with model misspecification. In *6th Annual Learning for Dynamics & Control Conference*, pages 980–992. PMLR, 2024a.

Bruce D Lee, Leonardo F Toso, Thomas T Zhang, James Anderson, and Nikolai Matni. Regret analysis of multi-task representation learning for linear-quadratic adaptive control. *arXiv preprint arXiv:2407.05781*, 2024b.

Xiang Li, Kaixuan Huang, Wenhao Yang, Shusen Wang, and Zhihua Zhang. On the convergence of fedavg on non-iid data. *arXiv preprint arXiv:1907.02189*, 2019.

Chaoyue Liu, Libin Zhu, and Mikhail Belkin. Loss landscapes and optimization in over-parameterized non-linear systems and neural networks. *Applied and Computational Harmonic Analysis*, 59:85–116, 2022.

Hao Liu, Richard Socher, and Caiming Xiong. Taming maml: Efficient unbiased meta-reinforcement learning. In *International conference on machine learning*, pages 4061–4071. PMLR, 2019.

Ricardo Luna Gutierrez and Matteo Leonetti. Information-theoretic task selection for meta-reinforcement learning. *Advances in Neural Information Processing Systems*, 33:20532–20542, 2020.

Dhruv Malik, Ashwin Pananjady, Kush Bhatia, Koulik Khamaru, Peter Bartlett, and Martin Wainwright. Derivative-free methods for policy optimization: Guarantees for linear quadratic systems. In *The 22nd international conference on artificial intelligence and statistics*, pages 2916–2925. PMLR, 2019.

Baharan Mirzasoleiman, Jeff Bilmes, and Jure Leskovec. Coresets for data-efficient training of machine learning models. In *International Conference on Machine Learning*, pages 6950–6960. PMLR, 2020.

Igor Molybog and Javad Lavaei. When does maml objective have benign landscape? In *2021 IEEE Conference on Control Technology and Applications (CCTA)*, pages 220–227. IEEE, 2021.

Negin Musavi and Geir E Dullerud. Convergence of Gradient-based MAML in LQR. *arXiv preprint arXiv:2309.06588*, 2023.

Anusha Nagabandi, Ignasi Clavera, Simin Liu, Ronald S Fearing, Pieter Abbeel, Sergey Levine, and Chelsea Finn. Learning to adapt in dynamic, real-world environments through meta-reinforcement learning. *arXiv preprint arXiv:1803.11347*, 2018.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. An analysis of approximations for maximizing submodular set functions. *Mathematical programming*, 14(1):265–294, 1978.

Yurii Nesterov and Vladimir Spokoiny. Random gradient-free minimization of convex functions. *Foundations of Computational Mathematics*, 17:527–566, 2017.

Samet Oymak and Mahdi Soltanolkotabi. Overparameterized nonlinear learning: Gradient descent takes the shortest path? In *International Conference on Machine Learning*, pages 4951–4960. PMLR, 2019.

Yunian Pan and Quanyan Zhu. Model-agnostic zeroth-order policy optimization for meta-learning of ergodic linear quadratic regulators. *arXiv preprint arXiv:2405.17370*, 2024.

Omead Pooladzandi, David Davini, and Baharan Mirzasoleiman. Adaptive second order coresets for data-efficient machine learning. In *International Conference on Machine Learning*, pages 17848–17869. PMLR, 2022.

Jonas Rothfuss, Dennis Lee, Ignasi Clavera, Tamim Asfour, and Pieter Abbeel. Promp: Proximal meta-policy search. *arXiv preprint arXiv:1810.06784*, 2018.

Tim Salimans, Jonathan Ho, Xi Chen, Szymon Sidor, and Ilya Sutskever. Evolution strategies as a scalable alternative to reinforcement learning. *arXiv preprint arXiv:1703.03864*, 2017.

Jaeuk Shin, Giho Kim, Howon Lee, Joonho Han, and Insoon Yang. On task-relevant loss functions in meta-reinforcement learning and online lqr. *arXiv preprint arXiv:2312.05465*, 2023.

Xingyou Song, Wenbo Gao, Yuxiang Yang, Krzysztof Choromanski, Aldo Pacchiano, and Yunhao Tang. Es-maml: Simple hessian-free meta learning. *arXiv preprint arXiv:1910.01215*, 2019.

Xingyou Song, Yuxiang Yang, Krzysztof Choromanski, Ken Caluwaerts, Wenbo Gao, Chelsea Finn, and Jie Tan. Rapidly adaptable legged robots via evolutionary meta-learning. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3769–3776. IEEE, 2020.

James C Spall. *Introduction to stochastic search and optimization: estimation, simulation, and control*. John Wiley & Sons, 2005.

Charles Stein. A bound for the error in the normal approximation to the distribution of a sum of dependent random variables. In *Proceedings of the Sixth Berkeley Symposium on Mathematical Statistics and Probability, Volume 2: Probability Theory*, volume 6, pages 583–603. University of California Press, 1972.

Sebastian U Stich. Local sgd converges fast and communicates little. *arXiv preprint arXiv:1805.09767*, 2018.

Yujie Tang, Zhaolin Ren, and Na Li. Zeroth-order feedback optimization for cooperative multi-agent systems. *Automatica*, 148:110741, 2023.

Rishabh Tiwari, Krishnateja Killamsetty, Rishabh Iyer, and Pradeep Shenoy. Gcr: Gradient coreset based replay buffer selection for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 99–108, 2022.

Leonardo F. Toso, Han Wang, and James Anderson. Learning Personalized Models with Clustered System Identification. *arXiv preprint arXiv:2304.01395*, 2023a.

Leonardo F. Toso, Donglin Zang, James Anderson, and Han Wang. Meta-Learning Linear Quadratic Regulators: A Policy Gradient MAML Approach for the Model-free LQR. 2023b. URL https://github.com/jd-anderson/MAML-LQR/blob/main/MAML_LQR_technical_report.pdf.

Leonardo F Toso, Han Wang, and James Anderson. Asynchronous Heterogeneous Linear Quadratic Regulator Design. *arXiv preprint arXiv:2404.09061*, 2024a.

Leonardo Felipe Toso, Donglin Zhan, James Anderson, and Han Wang. Meta-Learning Linear Quadratic Regulators: A Policy Gradient MAML Approach for Model-free LQR. 242:902–915, 15–17 Jul 2024b.

Mark Towers, Ariel Kwiatkowski, Jordan Terry, John U Balis, Gianluca De Cola, Tristan Deleu, Manuel Goulao, Andreas Kallinteris, Markus Krimmel, Arjun KG, et al. Gymnasium: A standard interface for reinforcement learning environments. *arXiv preprint arXiv:2407.17032*, 2024.

Joel A Tropp. User-friendly tail bounds for sums of random matrices. *Foundations of computational mathematics*, 12:389–434, 2012.

Han Wang, Leonardo F. Toso, Aritra Mitra, and James Anderson. Model-free Learning with Heterogeneous Dynamical Systems: A Federated LQR Approach. *arXiv preprint arXiv:2308.11743*, 2023.

Jane X Wang, Zeb Kurth-Nelson, Dhruva Tirumala, Hubert Soyer, Joel Z Leibo, Remi Munos, Charles Blundell, Dharshan Kumaran, and Matt Botvinick. Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*, 2016.

Zhenyi Wang, Li Shen, Tiehang Duan, Donglin Zhan, Le Fang, and Mingchen Gao. Learning to learn and remember super long multi-domain task sequence. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 7982–7992, 2022.

Laurence A Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2(4):385–393, 1982.

Yu Yang, Hao Kang, and Baharan Mirzasoleiman. Towards sustainable learning: Coresets for data-efficient deep learning. *arXiv preprint arXiv:2306.01244*, 2023.

Wenhao Yu, Jie Tan, C Karen Liu, and Greg Turk. Preparing for the unknown: Learning a universal policy with online system identification. *arXiv preprint arXiv:1702.02453*, 2017.

Wenhao Yu, Jie Tan, Yunfei Bai, Erwin Coumans, and Sehoon Ha. Learning fast adaptation with meta strategy optimization. *IEEE Robotics and Automation Letters*, 5(2):2950–2957, 2020.

Donglin Zhan and James Anderson. Data-efficient and robust task selection for meta-learning. *arXiv preprint arXiv:2405.07083*, 2024.

Thomas T Zhang, Katie Kang, Bruce D Lee, Claire Tomlin, Sergey Levine, Stephen Tu, and Nikolai Matni. Multi-task imitation learning for linear dynamical systems. In *Learning for Dynamics and Control Conference*, pages 586–599. PMLR, 2023.

6. Appendix

Roadmap: This appendix is organized as follows: First we extend our related work section and remind the reader of the model-agnostic meta-reinforcement learning problem and the matrix Bernstein inequality from Tropp (2012), where the later is crucial for controlling the error in the MAML gradient approximation due to the zeroth-order estimation. Next, in Section 6.3, we characterize that estimation error and prove that for a sufficiently large number of samples n_s, and sufficiently small smoothing radius r, the estimation error is composed of a sufficiently small error ϵ and an additive bias due to the task selection step in Algorithm 1, with high probability. Then, in Section 6.4 we derive the ergodic convergence rate of Algorithm 1 for the general setting of non-concave task-specific reward function $J_i(\theta)$. The sample complexity reduction benefit of task selection is then discussed in Section 6.5. In Sections 6.6, 6.7 and 6.8, we apply Algorithm 1 to the MAML-LQR problem where $J_i(\theta)$ satisfy a gradient dominance property. Finally in Section 6.9 we provide further details on the experimental setup considered in our numerical validation section.

6.1. Related Work

- **Meta-Reinforcement Learning:** There is a wealth of literature in meta-RL, with applications spanning robot manipulation (Yu et al., 2017; Arndt et al., 2020; Ghadirzadeh et al., 2021), robot locomotion (Song et al., 2020; Yu et al., 2020), build energy control (Luna Gutierrez and Leonetti, 2020), among others. Most relevant to our work are Song et al. (2019, 2020) that estimate the meta-gradient through evolutionary strategy. Similarly, we consider a zeroth-order estimation of the task-specific and meta-gradients. In contrast, these works treat all the tasks equally, leading to task redundancy which we handle with a derivative-free coreset learning approach to enhance data-efficiency in meta-RL.
- **Meta-Reinforcement Learning Task Selection:** Beyond the line of work on coresets for data-efficient training of machine learning models (Mirzasoleiman et al., 2020; Killamsetty et al., 2021; Yang et al., 2023; Pooladzandi et al., 2022; Balakrishnan et al., 2022), which use submodular optimization for subset selection, the works (Luna Gutierrez and Leonetti, 2020; Zhan and Anderson, 2024) are particularly relevant to this paper. In particular, Zhan and Anderson (2024) does not focus on RL tasks and approximates gradients using the pre-activation outputs of the last layer for classification tasks. That simplifies the problem but prevent them from deriving sample complexity guarantees. On the other hand, Shin et al. (2023) employs an information-theoretic metric to evaluate task similarities and relevance, considering a general MAML training framework rather than the policy gradient-based approach discussed here.
- **Model-free Learning for Control:** The linear quadratic regulator (LQR) problem has recently been taken as a fundamental baseline for establishing theoretical guarantees of policy optimization in control and reinforcement learning (Fazel et al., 2018). In particular, studies on multi-task and multi-agent learning for control (Zhang et al., 2023; Wang et al., 2023; Tang et al., 2023; Toso et al., 2024a,b; Lee et al., 2024a,b) have derived non-asymptotic guarantees for various learning architectures within the scope of model-free LQR. Most relevant to our work are Molybog and Lavaei (2021); Musavi and Dullerud (2023); Toso et al. (2024b); Aravind et al. (2024); Pan and Zhu (2024), which also study the meta-LQR problem and provide provable methods for learning meta-controllers that adapt quickly to unseen LQR tasks. In contrast to these works, we leverage the MAML-LQR problem as a case study to highlight the sample complexity reduction enabled by our embedded task selection approach.

6.2. Notation and Background Results

Notation: Let $[M]$ denote the set of integers $\{1, 2, \ldots, M\}$, and $\rho(\cdot)$ the spectral radius of a square matrix. Let $\|\cdot\|$ and $\|\cdot\|_F$ denote the spectral and Frobenius norm, respectively. We use $\mathcal{O}(\cdot)$ to omit constant factors in the argument. Throughout the text and when its clear from the context we use i and j to denote tasks T_i and T_j.

Model-Agnostic Meta-Reinforcement Learning Problem: We recall that the one-shot model-agnostic meta-reinforcement learning problem can be written as follows:

$$\theta^\star := \text{argmax}_{\theta\in\Theta} J(\theta) := \mathbb{E}_{i\sim\mathcal{P}(\mathcal{T})} J_i(\theta + \eta_{\text{inn}} \nabla J_i(\theta)),$$

where $J_i(\theta) := \mathbb{E}_{\tau\sim\mathcal{P}_i(\tau|\theta)}[\mathcal{R}_i(\tau)]$, and η_{inn} denotes some positive step-size.

Lemma 6 (Tropp (2012)) *(Matrix Bernstein Inequality) Let $\{Z_l\}_{l=1}^m$ be a set of m independent random matrices of dimension $d = d_1 \times d_2$ with $\mathbb{E}[Z_l] = Z$, $\|Z_l - Z\| \leq b$ almost surely, and maximum variance*

$$\max\left(\left\|\mathbb{E}\left(Z_l Z_l^\top\right) - ZZ^\top\right\|, \left\|\mathbb{E}\left(Z_l^\top Z_l\right) - Z^\top Z\right\|\right) \leq \sigma^2,$$

and sample average $\widehat{Z} := \frac{1}{m}\sum_{l=1}^m Z_l$. Let a small tolerance $\epsilon \geq 0$ and small probability $0 \leq \delta \leq 1$ be given. If

$$m \geq 2\left(\frac{\sigma^2}{\epsilon^2} + \frac{b}{3\epsilon}\right)\log\left(\frac{d_1 + d_2}{\delta}\right)$$

then $\mathbb{P}\left[\|\widehat{Z} - Z\| \leq \epsilon\right] \geq 1 - \delta$.

Lemma 7 (Young's inequality) *Given any two matrices $A, B \in \mathbb{R}^{d_1\times d_2}$ and a positive scalar ν, it holds that*

$$\|A + B\|_2^2 \leq (1+\nu)\|A\|_2^2 + \left(1 + \frac{1}{\nu}\right)\|B\|_2^2 \leq (1+\nu)\|A\|_F^2 + \left(1 + \frac{1}{\nu}\right)\|B\|_F^2, \tag{9a}$$

$$\langle A, B\rangle \leq \frac{\nu}{2}\|A\|_2^2 + \frac{1}{2\nu}\|B\|_2^2 \leq \frac{\nu}{2}\|A\|_F^2 + \frac{1}{2\nu}\|B\|_F^2. \tag{9b}$$

6.3. Gradient Estimation Error

Let $\tilde{J}(\theta) := \mathbb{E}_{u\sim\mathbb{S}_r}[J(\theta + u)]$ denote the Gaussian smoothing of the MAML reward function, with smoothing radius $r > 0$ and u being randomly drawn from a uniform distribution over matrices of dimension $d = d_1 \times d_2$ and operator norm r, $\mathbb{S}_r$. By using the two-point zeroth-order estimation we can define the gradient of the smoothed MAML reward function as

$$\nabla\tilde{J}(\theta) := \mathbb{E}_{j,u}\left[\frac{d}{2r^2}\left(J_j(\theta + u + \eta_{\text{inn}}\nabla\tilde{J}_j(\theta+u)) - J_j(\theta - u + \eta_{\text{inn}}\nabla\tilde{J}_j(\theta - u))\right)u\right],$$

where $\tilde{J}_j(\theta \pm u) = \mathbb{E}_{v\sim\mathbb{S}_r}[J_j(\theta \pm u + v)]$ denotes the Gaussian smoothing of the i-th task-specific expected reward function $J_j(\theta \pm u)$ incurred by the policy $\pi_{\theta\pm u}$. Note that the sample mean of $\nabla\tilde{J}(\theta)$ over tasks $i \sim \mathcal{P}(\mathcal{T})$ and samples $u \sim \mathbb{S}_r$ can be written as

$$\widetilde{\nabla} J(\theta) := \frac{d}{2Mn_s r^2} \sum_{j\in\mathcal{M}} \sum_{l=1}^{n_s} \Big(J_j(\theta + u_l + \eta_{\text{inn}} \nabla \tilde{J}_j(\theta + u_l)) - J_j(\theta - u_l + \eta_{\text{inn}} \nabla \tilde{J}_j(\theta - u_l)) \Big)\, u_l.$$

In addition, we define the two-point zeroth-order estimation of $\nabla J(\theta)$, over $\mathcal{M}$, as follows:

$$\nabla_{\mathcal{M}} J(\theta) := \frac{d}{2Mn_s r^2} \sum_{j\in\mathcal{M}} \sum_{l=1}^{n_s} \Big(J_j(\theta + u_l + \eta_{\text{inn}} \widehat{\nabla} J_j(\theta)) - J_j(\theta - u_l + \eta_{\text{inn}} \widehat{\nabla} J_j(\theta)) \Big)\, u_l$$

where

$$\widehat{\nabla} J_j(\theta) := \frac{d}{2n_s r^2} \sum_{l=1}^{n_s} \left(J_j(\theta + v_l) - J_j(\theta - v_l) \right) v_l,$$

and similarly we can define the estimation over the subset of tasks $\mathcal{S}$ as

$$\nabla_{\mathcal{S}} J(\theta) := \frac{d}{2Mn_s r^2} \sum_{i\in\mathcal{S}} \gamma_i \sum_{l=1}^{n_s} \Big(J_i(\theta + u_l + \eta_{\text{inn}} \widehat{\nabla} J_i(\theta)) - J_i(\theta - u_l + \eta_{\text{inn}} \widehat{\nabla} J_i(\theta)) \Big)\, u_l,$$

and define $g_i(\theta) = \frac{d}{2r^2 n_s} \sum_{l=1}^{n_s} \Big(J_i(\theta + u_l + \eta_{\text{inn}} \widehat{\nabla} J_i(\theta)) - J_i(\theta - u_l + \eta_{\text{inn}} \widehat{\nabla} J_i(\theta)) \Big)\, u_l$.

Goal: We aim to demonstrate that the error in approximation of the gradient over the subset of tasks $\mathcal{S}$, i.e., $\|\nabla J(\theta) - \nabla_{\mathcal{S}} J(\theta)\|$, is sufficient small if the number of samples n_s, the smoothing radius r and the step-size η_{inn} are set accordingly. To do so, let us first write the following

$$\begin{aligned} \|\nabla J(\theta) - \nabla_{\mathcal{S}} J(\theta)\| &= \|\nabla J(\theta) - \nabla_{\mathcal{M}} J(\theta) + \nabla_{\mathcal{M}} J(\theta) - \nabla_{\mathcal{S}} J(\theta)\| \\ &\le \underbrace{\|\nabla J(\theta) - \nabla_{\mathcal{M}} J(\theta)\|}_{\text{Zeroth-order gradient approximation}} + \underbrace{\|\nabla_{\mathcal{M}} J(\theta) - \nabla_{\mathcal{S}} J(\theta)\|}_{\text{Task selection}}, \end{aligned}$$

where we need to control the error in the gradient approximation over all the tasks in $\mathcal{M}$, and the error due to the task selection and training over the tasks in the subset $\mathcal{S}$.

• Zeroth-order gradient approximation $\|\nabla J(\theta) - \nabla_{\mathcal{M}} J(\theta)\|$:

$$\|\nabla J(\theta) - \nabla_{\mathcal{M}} J(\theta)\| \le \underbrace{\|\nabla J(\theta) - \nabla \tilde{J}(\theta)\|}_{\textbf{(a)}} + \underbrace{\|\nabla \tilde{J}(\theta) - \widetilde{\nabla} J(\theta)\|}_{\textbf{(b)}} + \underbrace{\|\widetilde{\nabla} J(\theta) - \nabla_{\mathcal{M}} J(\theta)\|}_{\textbf{(c)}},$$

(a):

$$\|\nabla J(\theta) - \nabla \tilde{J}(\theta)\| = \|\mathbb{E}_{u\in\mathbb{S}_r} \left(\nabla J(\theta) - \nabla J(\theta + u) \right) \|$$

$$
\begin{aligned}
&\overset{(i)}{\leq} \mathbb{E}_{j,u}\|\nabla J_j(\theta + \eta_{\text{inn}}\nabla J_j(\theta)) - \nabla J_j(\theta + u + \eta_{\text{inn}}\nabla J_j(\theta + u))\| \\
&\overset{(ii)}{\leq} \mathbb{E}_{j,u}\psi\|\eta_{\text{inn}}\left(\nabla J_j(\theta) - \nabla J_j(\theta + u)\right) - u\| \\
&\leq \psi r(1 + \psi\eta_{\text{inn}}) \overset{(iii)}{\leq} \frac{3\psi r}{2},
\end{aligned}
$$

where (i) and (ii) follows from Jensen's inequality and (6), respectively. Moreover, (iii) follows from selecting the step-size according to $\eta_{\text{inn}} \leq \frac{1}{2\psi}$. Therefore, by selecting the smoothing radius as $r \leq \frac{2\epsilon}{9\psi}$, we obtain $\|\nabla J(\theta) - \nabla \tilde{J}(\theta)\| \leq \frac{\epsilon}{3}$.

(b): we first note that $\|\nabla \tilde{J}(\theta) - \widetilde{\nabla} J(\theta)\| = \|\mathbb{E}_{j,u}\widetilde{\nabla} J(\theta) - \widetilde{\nabla} J(\theta)\|$. In addition, since the tasks $T_j \in \mathcal{M}$ and samples $u \sim \mathbb{S}_r$ are drawn independently, we can use Lemma 6 to control $\|\nabla \tilde{J}(\theta) - \widetilde{\nabla} J(\theta)\|$. Let us first denote $Z_l = \frac{d}{2r^2}\Big(J_j(\theta + u_l + \eta_{\text{inn}}\nabla \tilde{J}_j(\theta + u_l)) - J_j(\theta - u_l + \eta_{\text{inn}}\nabla \tilde{J}_j(\theta - u_l))\Big) u_l$ and $Z = \mathbb{E}_{j,u}[Z_l]$.

$$
\begin{aligned}
\|Z_l\| &\leq \frac{d}{2r}|J_j(\theta + u_l + \eta_{\text{inn}}\nabla \tilde{J}_j(\theta + u_l)) - J_j(\theta - u_l + \eta_{\text{inn}}\nabla \tilde{J}_j(\theta - u_l))| \\
&\overset{(i)}{\leq} \frac{d\beta J_{\max}}{2r}\|2u_l + \eta_{\text{inn}}\Big(\nabla \tilde{J}_j(\theta + u_l) - \nabla \tilde{J}_j(\theta - u_l)\Big)\| \\
&\leq d\beta J_{\max} + \frac{d\beta J_{\max}\eta_{\text{inn}}}{2r}\|\mathbb{E}_{v\sim\mathbb{S}_r}\nabla J_j(\theta + u_l + v) - \nabla J_j(\theta - u_l + v)\| \\
&\overset{(ii)}{\leq} d\beta J_{\max} + \frac{d\beta J_{\max}\eta_{\text{inn}}\psi}{2r}\|2u_j\| \leq \frac{3d\beta J_{\max}}{2},
\end{aligned}
$$

where (i) is due to (6) and $J_{\max} = \max_{j\in\mathcal{M},\theta\in\Theta} J_j(\theta)$. In addition, (ii) follows from the definition of the Gaussian smoothing of the task-specific reward function and from (6). The last inequality is due to $\eta_{\text{inn}} \leq \frac{1}{2\psi}$. Let us now control the approximation bias $\|Z_l - Z\|$.

$$
\|Z\| = \|\nabla \tilde{J}(\theta)\| \leq \frac{\epsilon}{3} + \|\nabla J(\theta)\| \leq \frac{\epsilon}{3} + \phi,
$$

which implies $\|Z_l - Z\| \leq \|Z_l\| + \|Z\| \leq b := \frac{3d\beta J_{\max}}{2} + \frac{\epsilon}{3} + \phi$. We control the approximation variance $\|\mathbb{E}(Z_l Z_l^\top) - ZZ^\top\|$ as follows:

$$
\begin{aligned}
\|\mathbb{E}(Z_l Z_l^\top) - ZZ^\top\| &\leq \|\mathbb{E}(Z_l Z_l^\top)\| + \|ZZ^\top\| \leq \max_{Z_l}(\|Z_l\|)^2 + \|Z\|^2 \\
&\leq \sigma^2 := \left(\frac{3d\beta J_{\max}}{2}\right)^2 + \left(\frac{\epsilon}{3} + \phi\right)^2,
\end{aligned}
$$

and by selecting $Mn_s \geq 36\left(\frac{\sigma^2}{\epsilon^2} + \frac{b}{\epsilon}\right)\log\left(\frac{d_1+d_2}{\delta}\right)$, we have that $\|\nabla \tilde{J}(\theta) - \widetilde{\nabla} J(\theta)\| \leq \frac{\epsilon}{3}$ holds with probability $1 - \delta$.

(c):

$$\begin{aligned}\|\widetilde{\nabla} J(\theta) - \nabla' J(\theta)\| &\leq \frac{d}{2Mn_s r^2} \sum_{j\in\mathcal{M}} \sum_{l=1}^{n_s} \| \Big(J_j(\theta + u_l + \eta_{\text{inn}} \nabla \tilde{J}_j(\theta + u_l)) - J_j(\theta + u_l + \eta_{\text{inn}} \widehat{\nabla} J_j(\theta)) \\ &\qquad + J_j(\theta - u_l + \eta_{\text{inn}} \widehat{\nabla} J_j(\theta)) - J_j(\theta - u_l + \eta_{\text{inn}} \nabla \tilde{J}_j(\theta - u_l)) \Big) u_l \| \\ &\overset{(i)}{\leq} \frac{d\beta J_{\max} \eta_{\text{inn}}}{2Mn_s r} \sum_{j\in\mathcal{M}} \sum_{l=1}^{n_s} \|\nabla \tilde{J}_j(\theta + u_l) - \widehat{\nabla} J_j(\theta)\| + \|\nabla \tilde{J}_j(\theta - u_l) - \widehat{\nabla} J_j(\theta)\| \\ &\overset{(ii)}{\leq} d\beta J_{\max} \psi \eta_{\text{inn}} + \frac{d\beta J_{\max} \eta_{\text{inn}}}{Mn_s r} \sum_{j\in\mathcal{M}} \sum_{l=1}^{n_s} \|\nabla \tilde{J}_j(\theta) - \widehat{\nabla} J_j(\theta)\|,\end{aligned}$$

where (i) is due to (6), and (ii) follows from adding and subtracting $\tilde{J}_j(\theta)$ and using (6) with the definition of Gaussian smoothing. Then, by (Flaxman et al., 2004, Lemma 1), $\nabla \tilde{J}_j(\theta) = \mathbb{E}_u[\frac{d}{r^2} J_j(\theta + u)u]$, which implies that $\nabla \tilde{J}_j(\theta) = \mathbb{E}\widehat{\nabla} J_j(\theta) = \mathbb{E}[\frac{d}{2r^2}(J_j(\theta + u) - J_j(\theta - u))\, u]$ due to the symmetric perturbation of the two-point zeroth-order approximation. Therefore, we proceed to control $\|\nabla \tilde{J}_j(\theta) - \widehat{\nabla} J_j(\theta)\| = \|\mathbb{E}_u \widehat{\nabla} J_j(\theta) - \widehat{\nabla} J_j(\theta)\|$, by using Lemma 6 as previously. In particular, we define $Z_l = \frac{d}{2r^2}(J_j(\theta + u_l) - J_j(\theta - u_l))\, u_l$ and $Z = \mathbb{E}_u[Z_l]$.

$$\|Z_k\| \leq \frac{d}{2r} |J_j(\theta + u_l) - J_j(\theta - u_l)| \overset{(i)}{\leq} \frac{d\beta J_{\max}}{2r} \|2u_l\| \leq d\beta J_{\max},$$

where (i) follows from (6). In addition, we have

$$\begin{aligned}\|Z\| = \|\nabla \tilde{J}_j(\theta)\| &\leq \|\nabla \tilde{J}_j(\theta) - \nabla J_j(\theta)\| + \phi \\ &\leq \mathbb{E}_u \|\nabla J_j(\theta + u) - \nabla J_j(\theta)\| + \phi \\ &\leq \psi r + \phi \leq \frac{\epsilon}{3} + \phi,\end{aligned}$$

where the last inequality follows from $r \leq \frac{\epsilon}{3\psi}$. Then, the approximation bias and variance of the estimation in **(c)** satisfy $\|Z_k - Z\| \leq b$ and $\|\mathbb{E}(Z_k Z_k^\top) - ZZ^\top\| \leq \sigma^2$, respectively. This implies that, by selecting $n_s \geq 18\left(\frac{4\sigma^2}{\epsilon^2} + \frac{b}{\epsilon}\right) \log\left(\frac{d_1 + d_2}{\delta}\right)$, $\|\nabla \tilde{J}_j(\theta) - \widehat{\nabla} J_j(\theta)\| \leq \frac{\epsilon}{6}$ holds with probability $1 - \delta$. Finally, by setting the inner step-size as $\eta_{\text{inn}} \leq \min\left\{\frac{r}{d\beta J_{\max}}, \frac{\epsilon}{6d\beta J_{\max}\psi}\right\}$, we have that $\|\widetilde{\nabla} J(\theta) - \nabla' J(\theta)\| \leq \frac{\epsilon}{3}$ holds with probability $1 - \delta$.

Therefore, by combining **(a)**, **(b)** and **(c)**, and supposing that the number of samples n_s, the smoothing radius r, and step-size η_{inn} are set as follows:

$$n_s \geq 36\left(\frac{8\sigma^2}{\epsilon^2} + \frac{b}{\epsilon}\right) \log\left(\frac{d}{\delta}\right), r \leq \frac{\epsilon}{9\psi}, \text{ and } \eta_{\text{inn}} \leq \min\left\{\frac{2r}{d\beta J_{\max}}, \frac{1}{2\psi}\right\},$$

the zeroth-order estimation error is sufficiently small, i.e., $\|\nabla J(\theta) - \nabla_{\mathcal{M}} J(\theta)\| \leq \epsilon$ holds with probability $1 - 2\delta$.

It is worth noting that the number of samples, smoothing radius and inner step-size must be in the order of $n_s = \mathcal{O}(d^2/\epsilon^2)$, $r = \mathcal{O}(\epsilon)$, and $\eta_{\text{inn}} = \mathcal{O}(\epsilon/d)$, respectively, in order to ensure that the estimation error due to the zeroth-order approximation is sufficiently small, i.e, $\|\nabla J(\theta) - \nabla_{\mathcal{M}} J(\theta)\| = \mathcal{O}(\epsilon)$. We use $\mathcal{O}(\cdot)$ to omit the dependence on universal constants and only highlight the scaling the number of samples with the problem dimension d and approximation error ϵ.

• Task Selection $\|\nabla_{\mathcal{M}} J(\theta) - \nabla_{\mathcal{S}} J(\theta)\|$: To control the estimation error due to the task selection, we start by writing the following

$$\begin{aligned}\|\nabla_{\mathcal{M}} J(\theta) - \nabla_{\mathcal{S}} J(\theta)\| &= \left\| \frac{1}{M} \sum_{j\in\mathcal{M}} g_j(\theta) - \frac{1}{M} \sum_{i\in\mathcal{S}} \gamma_i g_i(\theta) \right\| = \frac{1}{M} \left\| \sum_{j\in\mathcal{M}} g_j(\theta) - \sum_{i\in\mathcal{S}} \gamma_i g_i(\theta) \right\| \\ &\le \frac{1}{M} \sum_{j\in\mathcal{M}} \min_{i\in\mathcal{S}} \|g_j(\theta) - g_i(\theta)\| \le \frac{1}{M} \sum_{j\in\mathcal{M}} \min_{i\in\mathcal{S}} \max_{\theta\in\Theta} \|g_j(\theta) - g_i(\theta)\| .\end{aligned}$$

We recall that the greedy subset selection (i.e., steps 2-10 in Algorithm 1) returns a subset $\mathcal{S}$ that is a suboptimal solution of the following submodular maximization

$$\mathcal{S}^\star = \operatorname*{argmax}_{\mathcal{S}\subseteq\mathcal{M}} \ \mathcal{F}(\mathcal{S}) := C - \sum_{j\in\mathcal{M}} \min_{i\in\mathcal{S}} \|g_j(\theta_0) - g_i(\theta_0)\| \,, \text{ subject to } |\mathcal{S}| \le L,$$

for any $\theta_0 \in \Theta$. In particular, by (Nemhauser et al., 1978, Section 4), we know that the value of the greedy optimization is close to the optimal as $\mathcal{F}(\mathcal{S}) \ge (1 - e^{-1})\mathcal{F}(\mathcal{S}^\star)$. This implies that

$$\sum_{j\in\mathcal{M}} \min_{i\in\mathcal{S}} \|g_j(\theta_0) - g_i(\theta_0)\| \le Ce^{-1} + (1 - e^{-1}) \sum_{j\in\mathcal{M}} \min_{i\in\mathcal{S}^\star} \|g_j(\theta_0) - g_i(\theta_0)\| \,,$$

and by taking the maximum of both sides with respect to $\theta \in \Theta$, we obtain

$$\frac{1}{M} \sum_{j\in\mathcal{M}} \min_{i\in\mathcal{S}} \max_{\theta\in\Theta} \|g_j(\theta) - g_i(\theta)\| \le \frac{Ce^{-1}}{M} + \frac{(1 - e^{-1})}{M} \sum_{j\in\mathcal{M}} \min_{i\in\mathcal{S}^\star} \max_{\theta\in\Theta} \|g_j(\theta) - g_i(\theta)\| \,,$$

where we control $\|g_j(\theta) - g_i(\theta)\|$ as follows

$$\begin{aligned}\|g_j(\theta) - g_i(\theta)\| &\le \|g_j(\theta) - \nabla J_j(\theta + \eta_{\text{inn}} \nabla J_j(\theta))\| + \|\nabla J_j(\theta + \eta_{\text{inn}} \nabla J_j(\theta)) - \nabla J_j(\theta)\| \\ &\quad + \|g_i(\theta) - \nabla J_i(\theta + \eta_{\text{inn}} \nabla J_i(\theta))\| + \|\nabla J_i(\theta + \eta_{\text{inn}} \nabla J_i(\theta)) - \nabla J_i(\theta)\| \\ &\quad + \|\nabla J_i(\theta) - \nabla J_j(\theta)\| \\ &\overset{(i)}{\le} \|g_j(\theta) - \nabla J_j(\theta + \eta_{\text{inn}} \nabla J_j(\theta))\| + \|g_i(\theta) - \nabla J_i(\theta + \eta_{\text{inn}} \nabla J_i(\theta))\| \\ &\quad + 2\eta_{\text{inn}} \psi\phi + \|\nabla J_i(\theta) - \nabla J_j(\theta)\| \\ &\overset{(ii)}{\le} \left\| g_j(\theta) - \nabla J_j(\theta + \eta_{\text{inn}} \widehat{\nabla} J_j(\theta)) \right\| + \left\| g_i(\theta) - \nabla J_i(\theta + \eta_{\text{inn}} \widehat{\nabla} J_i(\theta)) \right\| \\ &\quad + \eta_{\text{inn}} \psi \|\widehat{\nabla} J_j(\theta) - \nabla J_j(\theta)\| + \eta_{\text{inn}} \psi \|\widehat{\nabla} J_i(\theta) - \nabla J_i(\theta)\| \\ &\quad + 2\eta_{\text{inn}} \psi\phi + \|\nabla J_i(\theta) - \nabla J_j(\theta)\|,\end{aligned}$$

where (i) and (ii) are due to (1). Therefore, we note that for either inner and outer zeroth-order gradient approximations, i.e., $\left\| g_j(\theta) - \nabla J_j(\theta + \eta_{\text{inn}} \widehat{\nabla} J_j(\theta)) \right\|$ and $\|\widehat{\nabla} J_j(\theta) - \nabla J_j(\theta)\|$, respectively, the number of samples and smoothing radius can be set according to

$$n_s \geq C_{\text{approx},1} \left(\frac{\sigma^2}{\epsilon^2} + \frac{b}{3\epsilon} \right) \log \left(\frac{d_1 + d_2}{\delta} \right), r \leq \frac{\epsilon}{C_{\text{approx},1} \psi}$$

to guarantee a small estimation error of ϵ, with probability $1 - \delta$, where $C_{\text{approx},1}$ is some positive universal constant, and $\sigma^2 = (d\beta J_{\max})^2 + (\epsilon + \phi)^2$, $b = d\beta J_{\max} + \epsilon + \phi$. Therefore, following Definition 2, i.e., $\xi_{i,j} := \max_{\theta \in \Theta} \|\nabla J_i(\theta) - \nabla J_j(\theta)\|$, the estimation error due to the task selection is bounded as follows:

$$\|\nabla_{\mathcal{M}} J(\theta) - \nabla_{\mathcal{M}} J(\theta)\| \leq \epsilon + \frac{1}{M} \sum_{j \in \mathcal{M}} \min_{i \in \mathcal{S}^\star} \xi_{i,j},$$

which holds with high probability $1-\delta$ for n_s and r as above, and $C \leq \frac{M\epsilon\epsilon}{C_{\text{approx},1}}$. Finally, the gradient estimation error $\|\nabla J(\theta) - \nabla_{\mathcal{S}} J(\theta)\|$ in Algorithm 1, is controlled by a sufficiently small error that comes from the zeroth-order estimation and an additive bias due to the task selection. That is,

$$\|\nabla J(\theta) - \nabla_{\mathcal{S}} J(\theta)\| \leq \epsilon + \frac{1}{M} \sum_{j \in \mathcal{M}} \min_{i \in \mathcal{S}^\star} \xi_{i,j},$$

which holds with high probability for $n_s = \mathcal{O}(d^2/\epsilon^2)$, $r = \mathcal{O}(\epsilon)$, $\eta_{\text{inn}} = \mathcal{O}(\epsilon/d)$, and $C = \mathcal{O}(\epsilon)$. We emphasize that setting C sufficiently small, i.e., $C = \mathcal{O}(\epsilon)$ is standard in the literature of coresets for data-efficient machine-learning (Mirzasoleiman et al., 2020; Pooladzandi et al., 2022; Yang et al., 2023) and it guarantees that the gradient estimation error due to the subset selection is sufficiently small.

Remark 8 *(Expected Reward vs Empirical Reward) It is worth noting that our previous derivations assume that we have access to an oracle that provides the task-specific expected reward $J_j(\theta)$ incurred by any policy π_θ, with $\theta \in \Theta$. However, in practice, we often do not have access to the true distribution of trajectories conditioned on the policy, i.e., $\tau \sim \mathcal{P}_i(\tau|\theta)$, that is needed to compute $J_j(\theta)$. Therefore, one may approximate the expected reward $J_j(\theta)$ with an empirical reward $\hat{J}_j(\theta) := \frac{1}{n_\tau} \sum_{l=1}^{n_\tau} [\mathcal{R}_j(\tau_l)]$, where $\{\tau_l\}_{l=1}^{n_\tau}$ are the trajectories obtained by playing with π_θ, n_τ times. Note that, we can control the error between $J_j(\theta)$ and $\hat{J}_j(\theta)$ with n_τ. Then, since that error should enters the analysis of Algorithm 1 for either with or without task selection settings, we may assume the access to $J_j(\theta)$, for simplicity, but we stress that our results can be readily extended to the practical setting of empirical rewards by controlling such error with a sufficiently large n_τ.*

6.4. Proof of Theorem 1 (ergodic Convergence Rate)

Recall that the meta-policy parameter is updated as follows:

$$\theta_{n+1} = \theta_n + \eta_{\text{out}} \nabla_{\mathcal{S}} J(\theta_n) = \theta_n + \frac{\eta_{\text{out}}}{M} \sum_{i \in \mathcal{S}} \gamma_i g_i(\theta_n).$$

In addition, by using the definition of the meta-gradient and the gradient Lipschitz assumption (6), we have that

$$\begin{aligned}
\|\nabla J(\theta_1) - \nabla J(\theta_2)\| &= \|\mathbb{E}_i \nabla J_i(\theta_1 + \eta_{\text{inn}} \nabla J_i(\theta_1)) - \mathbb{E}_i \nabla J_i(\theta_2 + \eta_{\text{inn}} \nabla J_i(\theta_2))\| \\
&\overset{(i)}{\leq} \mathbb{E}_i \psi \|(\theta_1 - \theta_2) + \eta_{\text{inn}} \left(\nabla J_i(\theta_1) - \nabla J_i(\theta_2)\right)\| \\
&\leq \psi \|\theta_1 - \theta_2\| + \eta_{\text{inn}} \|\nabla J_i(\theta_1) - \nabla J_i(\theta_2)\| \\
&\overset{(ii)}{\leq} \psi(1 + \eta_{\text{inn}} \psi) \|\theta_1 - \theta_2\| \\
&\overset{(iii)}{\leq} \frac{3\psi}{2} \|\theta_1 - \theta_2\| = \overline{\psi} \|\theta_1 - \theta_2\|,
\end{aligned}$$

for any $\theta_1, \theta_2 \in \Theta$. Here, (i) and (ii) follows from (2), and (iii) is due to $\eta_{\text{inn}} \leq \frac{1}{4\psi}$. Therefore, the MAML reward function incurred by policy π_θ is $\overline{\psi}$-smooth and satisfy

$$\begin{aligned}
J(\theta_n) - J(\theta_{n+1}) &\leq \langle \nabla J(\theta_n), \theta_n - \theta_{n+1} \rangle + \frac{\overline{\psi}}{2} \|\theta_{n+1} - \theta_n\|^2 \\
&= \langle \nabla J(\theta_n), -\eta_{\text{out}} \nabla_{\mathcal{S}} J(\theta_n) + \eta_{\text{out}} \nabla J(\theta_n) - \eta_{\text{out}} \nabla J(\theta_n) \rangle + \frac{\eta_{\text{out}}^2 \overline{\psi}}{2} \|\nabla_{\mathcal{S}} J(\theta_n)\|^2 \\
&= -\eta_{\text{out}} \|\nabla J(\theta_n)\|^2 + \eta_{\text{out}} \langle \nabla J(\theta_n), \nabla J(\theta_n) - \nabla_{\mathcal{S}} J(\theta_n) \rangle + \frac{\eta_{\text{out}}^2 \overline{\psi}}{2} \|\nabla_{\mathcal{S}} J(\theta_n)\|^2 \\
&\overset{(i)}{\leq} -\frac{\eta_{\text{out}}}{2} \|\nabla J(\theta_n)\|^2 + \frac{\eta_{\text{out}}}{2} \|\nabla J(\theta_n) - \nabla_{\mathcal{S}} J(\theta_n)\|^2 + \frac{\eta_{\text{out}}^2 \overline{\psi}}{2} \|\nabla_{\mathcal{S}} J(\theta_n)\|^2 \\
&\overset{(ii)}{\leq} -\frac{\eta_{\text{out}}}{2} \|\nabla J(\theta_n)\|^2 + \frac{\eta_{\text{out}}}{2} \|\nabla J(\theta_n) - \nabla_{\mathcal{S}} J(\theta_n)\|^2 \\
&\quad + \eta_{\text{out}}^2 \overline{\psi} \left(\|\nabla J(\theta_n)\|^2 + \|\nabla J(\theta_n) - \nabla_{\mathcal{S}} J(\theta_n)\|^2 \right),
\end{aligned}$$

where (i) and (ii) follows from Young's inequalities (9b) and (9a), respectively. Then, by setting $\eta_{\text{out}} \leq \frac{1}{2\overline{\psi}}$ and re-arranging the terms, we can write

$$\frac{\eta_{\text{out}}}{4} \|\nabla J(\theta_n)\|_2^2 \leq J(\theta_{n+1}) - J(\theta_n) + \frac{3\eta_{\text{out}}}{4} \|\nabla J(\theta_n) - \nabla_{\mathcal{S}} J(\theta_n)\|^2,$$

which can be unrolled over the iterations $n = \{0, 1, \ldots, N-1\}$ to obtain

$$\frac{1}{N} \sum_{n=0}^{N-1} \|\nabla J(\theta_n)\|_2^2 \leq \frac{4\Delta_0}{\eta_{\text{out}} N} + 6 \left(\frac{1}{M} \sum_{j \in \mathcal{M}} \min_{i \in \mathcal{S}^\star} \xi_{i,j} \right)^2,$$

where we also use the fact that $J(\theta^\star) \geq J(\theta_N)$ above. In addition, we disregard $\mathcal{O}(\epsilon^2)$ since it is negligible for small ϵ. We also note that the first term denotes the local algorithm's complexity to find an stationary solution given an initial optimality gap $\Delta_0 = J(\theta^\star) - J(\theta_0)$, and the second term scales with the gradient approximation error due to the zeroth-order estimation and task selection. Therefore, by setting the number of iterations as $N = \mathcal{O}(1/\epsilon)$, Algorithm 1 satisfy

$$\frac{1}{N} \sum_{n=0}^{N-1} \|\nabla J(\theta_n)\|_2^2 \leq \mathcal{O} \left(\epsilon + \left(\frac{1}{M} \sum_{j \in \mathcal{M}} \min_{i \in \mathcal{S}^\star} \xi_{i,j} \right)^2 \right). \tag{10}$$

6.5. Proof of Corollary 1 (Sample Complexity)

We let $\mathcal{S}_c^{\mathcal{S}}$ and $\mathcal{S}_c^{\mathcal{M}}$ denote the total number of samples required in Algorithm 1 to find an ϵ-near stationary solution, with and without task selection, respectively. In particular, $\mathcal{S}_c^{\mathcal{S}} = \mathcal{O}\left(LNn_s\right) + \mathcal{O}\left(Mn_s\right)$ and $\mathcal{S}_c^{\mathcal{M}} = \mathcal{O}\left(MNn_s\right)$. Note that in order to guarantee (10) with high probability, we need $n_s = \mathcal{O}(d^2/\epsilon^2)$ samples in the zeroth-order gradient estimation. Therefore, we have that $\mathcal{S}_c^{\mathcal{M}} = \mathcal{O}\left(MNn_s\right) = \mathcal{O}(d^2/\epsilon^3)$, whereas $\mathcal{S}_c^{\mathcal{S}} = \frac{L}{M}\mathcal{O}\left(d^2/\epsilon^3\right) + \mathcal{O}\left(d^2/\epsilon^2\right)$. Then, for a sufficiently large amount of tasks in the task pool $\mathcal{M}$ (e.g., scaling as $M = \mathcal{O}(1/\epsilon)$) and a sufficiently small number of informative tasks in the subset $\mathcal{S}$, i.e., $L = \mathcal{O}(1)$, the task selection benefits from a sample complexity reduction by a factor of $\mathcal{O}(1/\epsilon)$ when compared to the setting without task selection.

6.6. Proof of Theorem 2 (Optimality Gap)

We first note Lemma 5 is provided in the Frobenius norm. Therefore, we use the fact that for any matrix $A \in \mathbb{R}^{d_1 \times d_2}$, $\|A\|_F \leq \sqrt{\min(d_1, d_2)}\|A\|$, to adapt the gradient approximation error $\|\nabla J(\theta) - \nabla_{\mathcal{S}} J(\theta)\|$ as discussed previously in the RL setting, for the MAML-LQR where $\theta = K$ and $u \sim \mathbb{S}_r$ has Frobenius norm r in $\texttt{ZO2P}(\cdot)$. Moreover, we recall that the meta-controller is updated as follows:

$$K_{n+1} = K_n - \eta_{\text{out}} \nabla_{\mathcal{S}} J(K_n) = \theta_n - \frac{\eta_{\text{out}}}{M} \sum_{i \in \mathcal{S}} \gamma_i g_i(K_n),$$

where $g_i(K_n) = \frac{d}{2r^2 n_s} \sum_{l=1}^{n_s} \left(J_i(K_n + u_l - \eta_{\text{inn}} \widehat{\nabla} J_i(K_n)) - J_i(K_n - u_l - \eta_{\text{inn}} \widehat{\nabla} J_i(K_n)) \right) u_l$ for any task $T_i \sim \mathcal{P}(\mathcal{T})$. Then, by the gradient smoothness property in Lemma 5, we can write

$$\begin{aligned}
J_j(K_{n+1}) - J_j(K_n) &\leq \langle \nabla J_j(K_n), K_{n+1} - K_n \rangle + \frac{\psi}{2} \|K_{n+1} - K_n\|_F^2 \\
&= \langle \nabla J_j(K_n), -\eta_{\text{out}} \nabla_{\mathcal{S}} J(K_n) - \eta_{\text{out}} \nabla J_j(K_n) + \eta_{\text{out}} \nabla J_j(K_n) \rangle + \frac{\psi \eta_{\text{out}}^2}{2} \|\nabla_{\mathcal{S}} J(K_n)\|_F^2 \\
&\leq -\frac{\eta_{\text{out}}}{2} \|\nabla J_j(K_n)\|_F^2 + \frac{\eta}{2} \|\nabla_{\mathcal{S}} J(K_n) - \nabla J_j(K_n)\|_F^2 + \frac{\psi \eta_{\text{out}}^2}{2} \|\nabla_{\mathcal{S}} J(K_n)\|_F^2 \\
&\leq -\frac{\eta_{\text{out}}}{4} \|\nabla J_j(K_n)\|_F^2 + \frac{3\eta_{\text{out}}}{4} \|\nabla_{\mathcal{S}} J(K_n) - \nabla J_j(K_n)\|_F^2,
\end{aligned}$$

where the last two inequalities are due to Young's inequality (9b) and (9a), and $\eta_{\text{out}} \leq \frac{1}{4\psi}$. Let us now proceed to control the error in the meta-gradient approximation over $\mathcal{S}$, i.e., $\nabla_{\mathcal{S}} J(K_n)$, with respect to the task-specific gradient $\nabla J_j(K_n)$.

$$\|\nabla_{\mathcal{S}} J(K_n) - \nabla J_j(K_n)\|_F \leq \underbrace{\|\nabla_{\mathcal{S}} J(K_n) - \nabla J(K_n)\|_F}_{\text{Gradient approximation}} + \underbrace{\|\nabla J(K_n) - \nabla J_j(K_n)\|_F}_{\text{Task heterogeneity}},$$

- **Task heterogeneity:**

$$\|\nabla J(K_n) - \nabla J_j(K_n)\|_F = \|\mathbb{E}_i \nabla J_i(K_n - \eta_{\text{inn}} \nabla J_i(K_n)) - \nabla J_j(K_n)\|_F$$

$$
\begin{aligned}
&\leq \mathbb{E}_i \left\|\nabla J_i(K_n - \eta_{\text{inn}} \nabla J_i(K_n)) - \nabla J_i(K_n) + \nabla J_i(K_n) - \nabla J_j(K_n)\right\|_F \\
&\leq \mathbb{E}_i \left\|\nabla J_i(K_n - \eta_{\text{inn}} \nabla J_i(K_n)) - \nabla J_i(K_n)\right\|_F + \mathbb{E}_i \|\nabla J_i(K_n) - \nabla J_j(K_n)\|_F \\
&\overset{(i)}{\leq} \eta_{\text{inn}} \psi \mathbb{E}_i \|\nabla J_i(K_n)\| + \mathbb{E}_i \|\nabla J_i(K_n) - \nabla J_j(K_n)\|_F \\
&\overset{(ii)}{\leq} \eta_{\text{inn}} \psi \phi + f(\epsilon_{\text{het}}),
\end{aligned}
$$

where (i) is due to Lemma 5 and (ii) follows from Lemmas 4 and 5.

- **Gradient approximation:**

$$
\begin{aligned}
\|\nabla_{\mathcal{S}} J(K_n) - \nabla J(K_n)\|_F &= \|\nabla_{\mathcal{S}} J(K_n) - \nabla_{\mathcal{M}} J(K_n) + \nabla_{\mathcal{M}} J(K_n) - \nabla J(K_n)\|_F \\
&\leq \underbrace{\|\nabla_{\mathcal{S}} J(K_n) - \nabla_{\mathcal{M}} J(K_n)\|_F}_{\text{Task selection}} + \underbrace{\|\nabla_{\mathcal{M}} J(K_n) - \nabla J(K_n)\|_F}_{\text{Zeroth-order approximation}}.
\end{aligned}
$$

Following the previous analysis for the gradient estimation error of Algorithm 1 in Section 6.3, we know that $\|\nabla_{\mathcal{S}} J(K_n) - \nabla J(K_n)\|_F$ satisfy

$$
\|\nabla_{\mathcal{S}} J(K_n) - \nabla J(K_n)\|_F \leq \epsilon + \frac{1}{M} \sum_{j \in \mathcal{M}} \min_{i \in \mathcal{S}^\star} \xi_{i,j},
$$

with probability $1 - \delta$, if the number of samples and smoothing radius are selected according to

$$
n_s \geq C_{\text{approx,2}} \min(d_1, d_2) \left(\frac{\sigma^2}{\epsilon^2} + \frac{b}{3\sqrt{\min(d_1, d_2)}\epsilon} \right) \log\left(\frac{d_1 + d_2}{\delta} \right), r \leq \frac{\epsilon}{C_{\text{approx,2}} \psi}, \tag{11}
$$

with $\eta_{\text{inn}} = \mathcal{O}(\epsilon)$ and $C = \mathcal{O}(\epsilon)$. We also note that $\min(d_1, d_2)$ and $\sqrt{\min(d_1, d_2)}$ come from the Frobenius norm in Lemma 5. Then, we can write

$$
\begin{aligned}
J_j(K_{n+1}) - J_j(K_n) &\leq -\frac{\eta_{\text{out}}}{4} \|\nabla J_j(K_n)\|_F^2 + 3\eta_{\text{out}} \left(\eta_{\text{inn}}^2 \psi^2 \phi^2 + f^2(\epsilon_{\text{het}}) + \epsilon^2 + \left(\frac{1}{M} \sum_{j \in \mathcal{M}} \min_{i \in \mathcal{S}^\star} \xi_{i,j} \right)^2 \right) \\
&\overset{(i)}{\leq} -\frac{\lambda_j \eta_{\text{out}}}{4} \left(J_j(K_n) - J_j(K_j^\star) \right) + 3\eta_{\text{out}} \left(f^2(\epsilon_{\text{het}}) + \left(\frac{1}{M} \sum_{j \in \mathcal{M}} \min_{i \in \mathcal{S}^\star} \xi_{i,j} \right)^2 \right) \\
&\overset{(ii)}{\leq} \frac{\lambda_j \eta_{\text{out}}}{4} \left(J_j(K_n) - J_j(K_j^\star) \right) + 6\eta_{\text{out}} f^2(\epsilon_{\text{het}}),
\end{aligned}
$$

where (i) follows from the gradient dominance property in Lemma 5, $\eta_{\text{inn}} = \mathcal{O}(\epsilon)$ and disregarding $\mathcal{O}(\epsilon^2)$ since it is negligible for small ϵ. (ii) is due to the fact that $\frac{1}{M} \sum_{j \in \mathcal{M}} \min_{i \in \mathcal{S}^\star} \xi_{i,j} \leq f(\epsilon_{\text{het}})$. Then, we can add and subtract $J_j(K_j^\star)$ on the LHS to obtain

$$
\Delta_{n+1}^{(j)} \leq \left(1 - \frac{\lambda_j \eta_{\text{out}}}{4} \right) \Delta_n^{(j)} + 6\eta_{\text{out}} f^2(\epsilon_{\text{het}}),
$$

where $\Delta_n^{(j)} = J_j(K_n) - J_j(K_j^\star)$. Therefore, by unrolling the above expression over the iterations, $n = \{0, 1, \dots, N-1\}$, we obtain

$$\Delta_N^{(j)} \leq \left(1 - \frac{\lambda_j \eta_{\text{out}}}{4}\right)^N \Delta_0^{(j)} + \frac{24}{\lambda_j} f^2(\epsilon_{\text{het}}) \leq \epsilon + \frac{24}{\lambda_j} f^2(\epsilon_{\text{het}}), \tag{12}$$

where the last inequality is due to $N \geq \frac{4}{\eta_{\text{out}}\lambda_j} \log\left(\frac{\Delta_0^{(j)}}{\epsilon}\right)$. Then, we can conclude that Algorithm 1, for the MAML-LQR problem, learns a meta-controller K_N that is ϵ-close to any task-specific optimal controller (i.e., $T_j \sim \mathcal{P}(\mathcal{T})$) up to a task heterogeneity bias that scales as $\mathcal{O}\left(f^2(\epsilon_{\text{het}})\right)$.

6.7. Proof of Corollary 2 (Sample Complexity)

We let $\bar{\mathcal{S}}_c^{\mathcal{S}}$ and $\bar{\mathcal{S}}_c^{\mathcal{M}}$ to denote the total number of samples required in Algorithm 1, for the MAML-LQR problem, to learn a meta-controller K_N that is ϵ-close to any task-specific optimal controller up to a heterogeneity bias, with and without task selection, respectively. In particular, $\bar{\mathcal{S}}_c^{\mathcal{S}} = \mathcal{O}\left(LNn_s\right) + \mathcal{O}\left(Mn_s\right)$ and $\bar{\mathcal{S}}_c^{\mathcal{M}} = \mathcal{O}\left(MNn_s\right)$. In addition, to guarantee (12) with high probability, we need $n_s = \mathcal{O}(d^2/\epsilon^2)$ samples in the zeroth-order gradient estimation. Therefore, $\bar{\mathcal{S}}_c^{\mathcal{M}} = \mathcal{O}\left(MNn_s\right) = \mathcal{O}((d^2/\epsilon^2)\log(1/\epsilon))$, whereas $\bar{\mathcal{S}}_c^{\mathcal{S}} = \frac{L}{M}\mathcal{O}\left((d^2/\epsilon^2)\log(1/\epsilon)\right) + \mathcal{O}\left(d^2/\epsilon^2\right)$, then, for a sufficiently large amount of tasks in the task pool $\mathcal{M}$ (e.g., scaling as $M = \mathcal{O}(\log(1/\epsilon))$) and a sufficiently small amount of tasks in the subset $\mathcal{S}$, i.e., $L = \mathcal{O}(1)$, the task selection may benefit from a sample complexity reduction of a factor of up to $\mathcal{O}(\log(1/\epsilon))$ when compared to the setting without task selection.

6.8. Stability Analysis

We now proceed to demonstrate that $K_n \in \mathcal{G}$, for any $n = \{0, 1, \dots, N-1\}$ of Algorithm 1. Let us first recall that by Assumption 4, the initial meta-controller is stabilizing, i.e., $K_0 \in \mathcal{G}$. Then, we can first show that $\bar{K}_0 = K_0 - \eta_{\text{inn}}\widehat{\nabla} J_j(K_0)$ and $K_1 = K_0 - \eta_{\text{out}}\nabla_{\mathcal{S}} J(K_0)$ never leaves $\mathcal{G}$, with high probability, if the number of samples n_s, smoothing radius r, inner and outer step-sizes η_{inn}, η_{out}, and heterogeneity $f(\epsilon_{\text{text}})$ are set accordingly. Finally, we can use an induction step to extend the same for any iteration n. By the gradient smoothness in Lemma 5 we can write

$$J_j(\bar{K}_0) - J_j(K_0) \leq -\frac{\eta_{\text{inn}}}{4}\|\nabla J_j(K_0)\|_F^2 + \frac{3\eta_{\text{inn}}}{4}\|\widehat{\nabla} J_j(K_0) - \nabla J_j(K_0)\|_F^2,$$

where $\eta_{\text{out}} \leq \frac{1}{4\psi}$. Then, by using the gradient dominance property we have that

$$J_j(\bar{K}_0) - J_j(K_j^\star) \leq \left(1 - \frac{\lambda_j \eta_{\text{inn}}}{4}\right)\Delta_0^{(j)} + \frac{3\eta_{\text{inn}}}{4}\|\widehat{\nabla} J_j(K_0) - \nabla J_j(K_0)\|_F^2,$$

where $\|\widehat{\nabla} J_j(K_0) - \nabla J_j(K_0)\|_F^2$ corresponds to the zeroth-order gradient estimation error at K_0. As well-established in (Toso et al., 2024b,a) and also discussed previously in this work, the zeroth-order estimation error can be made arbitrarily small, for instance, $\|\widehat{\nabla} J_j(K_0) - \nabla J_j(K_0)\|_F \leq \sqrt{\frac{\lambda_j \Delta_0^{(j)}}{6}}$, for $n_s = \mathcal{O}\left(\frac{6\psi}{\lambda_j}\right)$ and $r = \mathcal{O}\left(\sqrt{\frac{\lambda_j \Delta_0^{(j)}}{6}}\right)$. Then, for all tasks $T_j \sim \mathcal{P}(\mathcal{T})$ we have

$$J_j(\bar{K}_0) - J_j(K_j^\star) \leq \left(1 - \frac{\lambda_j \eta_{\text{inn}}}{8}\right)\Delta_0^{(j)},$$

which implies that $K_0 \in \mathcal{G}$ (i.e., see Definition 3). We proceed to show that $K_1 \in \mathcal{G}$. To do so, we use again the gradient smoothness in Lemma 5 to write

$$\begin{aligned}
J_j(K_1) - J_j(K_0) &\leq -\frac{\eta_{\text{out}}}{4}\|\nabla J_j(K_0)\|_F^2 + \frac{3\eta_{\text{out}}}{4}\|\nabla_{\mathcal{S}} J(K_0) - \nabla J_j(K_0)\|_F^2 \\
&\overset{(i)}{\leq} -\frac{\eta_{\text{out}}}{4}\|\nabla J_j(K_0)\|_F^2 + \frac{3\eta_{\text{out}}}{2}\|\nabla_{\mathcal{S}} J(K_0) - \nabla J(K_0)\|_F^2 + 3\eta_{\text{out}} f^2(\epsilon_{\text{het}}) + 3\eta_{\text{out}}\eta_{\text{inn}}^2\phi^2,
\end{aligned}$$

where (i) is due to Young's inequality (9a) and

$$\|\nabla J(K_0) - J_j(K_0)\|_F^2 \leq \mathbb{E}_i\|\nabla J_i(K_0 - \eta_{\text{inn}}\nabla J_i(K_0)) - \nabla J_j(K_0)\|_F^2 \leq 2\eta_{\text{inn}}^2\phi^2 + 2f^2(\epsilon_{\text{het}}).$$

Then, by the gradient dominance property we have that

$$\begin{aligned}
J_j(K_1) - J_j(K_j^\star) \leq \left(1 - \frac{\lambda_j\eta_{\text{out}}}{4}\right)\Delta_0^{(j)} + \frac{3\eta_{\text{out}}}{2}\|\nabla_{\mathcal{S}} J(K_0) - \nabla J(K_0)\|_F^2 \\
+ 3\eta_{\text{out}} f^2(\epsilon_{\text{het}}) + 3\eta_{\text{out}}\eta_{\text{inn}}^2\phi^2,
\end{aligned}$$

where the gradient estimation error $\|\nabla_{\mathcal{S}} J(K_0) - \nabla J(K_0)\|_F$ satisfy

$$\|\nabla_{\mathcal{S}} J(K_0) - \nabla J(K_0)\|_F \leq \zeta + \frac{1}{M}\sum_{j\in\mathcal{M}} \min_{i\in\mathcal{S}^\star} \xi_{i,j} \leq \zeta + f(\epsilon_{\text{het}}),$$

with probability $1-\delta$, for n_s and r satisfying (11) with ζ in lieu of ϵ. Then, we can write

$$\begin{aligned}
J_j(K_1) - J_j(K_j^\star) &\leq \left(1 - \frac{\lambda_j\eta_{\text{out}}}{4}\right)\Delta_0^{(j)} + 3\eta_{\text{out}}\zeta^2 + 6\eta_{\text{out}} f^2(\epsilon_{\text{het}}) + 3\eta_{\text{out}}\eta_{\text{inn}}^2\phi^2 \\
&\overset{(i)}{\leq} \left(1 - \frac{\lambda_j\eta_{\text{out}}}{8}\right)\Delta_0^{(j)}
\end{aligned}$$

where (i) follows from $\zeta = \sqrt{\frac{\lambda_j\Delta_0^{(j)}}{72}}$, $f(\epsilon_{\text{het}}) \leq \frac{\lambda_j\Delta_0^{(j)}}{72}$, and $\eta_{\text{inn}} \leq \sqrt{\frac{\lambda_j\Delta_0^{(j)}}{72\phi^2}}$, which implies that $K_1 \in \mathcal{G}$. Therefore, we define our base case and inductive hypothesis as follows:

$$\begin{aligned}
\textbf{Base case: } & J_j(\bar{K}_0) - J_j(K_j^\star) \leq J_j(K_0) - J_j(K_j^\star), \\
& J_j(K_1) - J_j(K_j^\star) \leq J_j(K_0) - J_j(K_j^\star)
\end{aligned}$$

$$\begin{aligned}
\textbf{Inductive hypothesis: } & J_j(\bar{K}_n) - J_j(K_j^\star) \leq J_j(K_0) - J_j(K_j^\star), \\
& J_j(K_n) - J_j(K_j^\star) \leq J_j(K_0) - J_j(K_j^\star)
\end{aligned}$$

which can be used along with the aforementioned conditions on the number of samples, smoothing radius, step-sizes and heterogeneity, to write

$$\begin{aligned}
J_j(K_{n+1}) - J_j(K_j^\star) &\leq \left(1 - \frac{\lambda_j\eta_{\text{out}}}{4}\right)\Delta_n^{(j)} + 3\eta_{\text{out}}\zeta^2 + 6\eta_{\text{out}} f^2(\epsilon_{\text{het}}) + 3\eta_{\text{out}}\eta_{\text{inn}}^2\phi^2 \\
&\overset{(i)}{\leq} \left(1 - \frac{\lambda_j\eta_{\text{out}}}{8}\right)\Delta_0^{(j)} \leq \Delta_0^{(j)},
\end{aligned}$$

which guarantees that Algorithm 1 produces MAML stabilizing controllers with high probability.

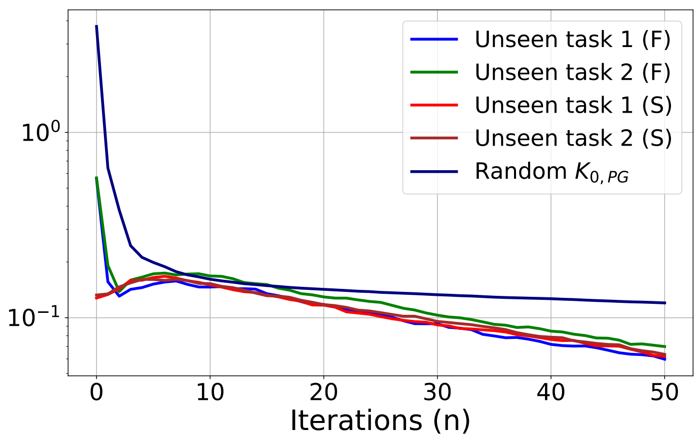

Figure 3: Task-specific optimality gap on unseen tasks with coreset selection (S), MAML-LQR (F), and random initialized controller $K_{0,\mathrm{PG}}$.

6.9. Numerical Validation

Reinforcement Learning - Cart Pole: In this experiment, we configured the cart pole environment with parameters uniformly sampled from predefined intervals: cart mass from [0.8, 1.2], pole mass from [0.08, 0.12], and pole length from [0.4, 0.6]. We maintained an episodic task pool of 800 tasks with a selection ratio of 25%. The learning rates were set to 0.2 for the inner loop and 0.05 for the meta-learning process. For each iteration, we used a batch size of 20 tasks. All experiments were implemented using PyTorch and OpenAI Gym (Towers et al., 2024), running on an NVIDIA GeForce 3090 GPU with 90GB RAM.

We compared the wall-clock running time between our proposed task selection algorithm and vanilla MAML. The average per-iteration running time was 3.32s for the task selection algorithm and 2.89s for vanilla MAML, demonstrating that our approach achieves significantly faster convergence while maintaining comparable computational efficiency.

Linear Quadratic Regulator (LQR): For the detailed experimental setup of the MAML-LQR setting, we strictly follow Section 4 of Toso et al. (2024b). We include the task-specific optimality gap for unseen tasks in this subsection as well. Figure 3 shows the task-specific optimality gap on unseen meta-testing tasks drawn from the same training task distribution $\mathcal{P}(\mathcal{T})$. The results show that the learned meta-controller on $\mathcal{S}$ achieves comparable generalization performance while reducing sample complexity in the meta-training, which outperforms the randomly initialized controller.