POMRL: No-Regret Learning-to-Plan with Increasing Horizons

Anonymous authors
Paper under double-blind review

Abstract

We study the problem of planning under model uncertainty in an online meta-reinforcement learning (RL) setting where an agent is presented with a sequence of related tasks with limited interactions per task. The agent can use its experience in each task *and* across tasks to estimate both the transition model and the distribution over tasks. We propose an algorithm to meta-learn the underlying structure across tasks, utilize it to plan in each task, and upper-bound the regret of the planning loss. Our bound suggests that the average regret over tasks decreases as the number of tasks increases and as the tasks are more similar. In the classical single-task setting, it is known that the planning horizon should depend on the estimated model's accuracy, that is, on the number of samples within task. We generalize this finding to meta-RL and study this dependence of planning horizons on the number of tasks. Based on our theoretical findings, we derive heuristics for selecting slowly increasing discount factors, and we validate its significance empirically.

1 Introduction

Meta-learning (Caruana, 1997; Baxter, 2000; Thrun & Pratt, 1998; Finn et al., 2017; Denevi et al., 2018) offers a powerful paradigm to leverage past experience to reduce the sample complexity of learning future related tasks. *Online meta-learning* considers a sequential setting, where the agent progressively accumulates knowledge and uses past experience to learn good priors and to quickly adapt within each task Finn et al. (2019); Denevi et al. (2019). When the tasks share a structure, such approaches enable progressively faster convergence, or equivalently better model accuracy with better sample complexity (Schmidhuber & Huber, 1991; Thrun & Pratt, 1998; Baxter, 2000; Finn et al., 2017; Balcan et al., 2019).

In model-based reinforcement learning (RL), the agent uses an estimated model of the environment to plan actions ahead towards the goal of maximizing rewards. A key component in the agent's decision making is the horizon used during planning. In general, an *evaluation horizon* is imposed by the task itself, but the learner may want to use a different and potentially shorter *guidance horizon*. In the discounted setting, the size of the evaluation horizon is of order $(1-\gamma_{\texttt{eval}})^{-1}$, for some discount factor $\gamma_{\texttt{eval}} \in (0,1)$, and the agent may use $\gamma \neq \gamma_{\texttt{eval}}$ for planning. For instance, a classic result known as Blackwell Optimality (Blackwell, 1962) states there exists a discount factor $\gamma^\star$ and a corresponding optimal policy such that the policy is also optimal for any greater discount factor $\gamma \geq \gamma^\star$. Thus, an agent that plans with $\gamma = \gamma^\star$ will be optimal for any $\gamma_{\texttt{eval}} > \gamma^\star$. In the Arcade Learning Environment (Bellemare et al., 2013) a discount factor of $\gamma_{\texttt{eval}} = 1$ is used for evaluation, but typically a smaller γ is used for training (Mnih et al., 2015). Using a smaller discount factor acts as a regularizer (Amit et al., 2020; Petrik & Scherrer, 2008; Van Seijen et al., 2009; François-Lavet et al., 2019; Arumugam et al., 2018) and reduces planner over-fitting in random MDPs (Arumugam et al., 2018). Indeed, the choice of planning horizon plays a significant role in computation (Kearns et al., 2002), optimality (Kocsis & Szepesvári, 2006), and on the complexity of the policy class (Jiang et al., 2015). In addition, meta-learning discount factors has led to significant improvements in performance (Xu et al., 2018; Zahavy et al., 2020; Flennerhag et al., 2021; 2022; Luketina et al., 2022).

When doing model-based RL with a learned model, the optimal guidance planning horizon, called *effective* horizon by Jiang et al. (2015), depends on the accuracy of the model, and so on the amount of data used to

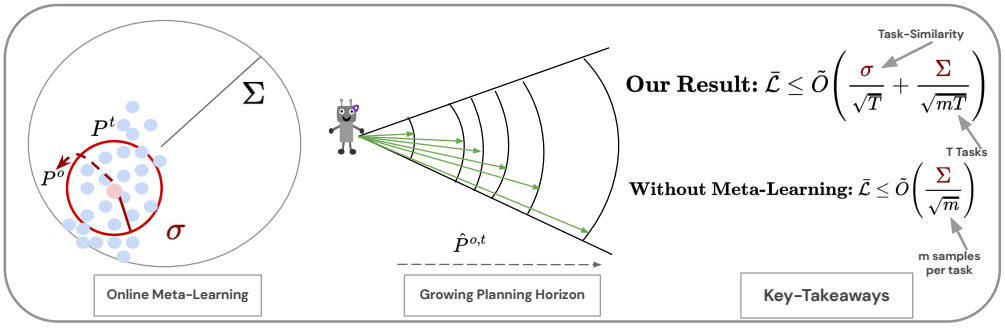


Figure 1: **Effective Planning Horizons in Meta-Reinforcement Learning.** The agent faces a sequence of tasks P^t whose optimal parameters are close to each other ($\sigma < \Sigma = 1$). The agent builds a transition model for each task and plans with these inaccurate models. By using data from previous tasks, the agent meta-learns an initialization of the model ($\hat{P}^{o,t}$), which leads to better planning in new related but unseen tasks. We show an improved average regret upper bound that scales with task-similarity σ and inversely with the number of tasks T: as knowledge accumulates, uncertainty diminishes, and the agent can plan with longer horizons.

estimate it. Jiang et al. (2015) show that when data is scarce, a guidance discount factor $\gamma < \gamma_{\texttt{eval}}$ should be preferred for planning. The reason for this is straightforward; if the model used for planning is inaccurate, then errors will tend to accumulate along the planned trajectory. A shorter effective planning horizon will accumulate less error and may lead to better performance, even when judged using the true $\gamma_{\texttt{eval}}$. While that work treated only the batch, single-task setting, the question of effective planning horizon remains open in the online meta-learning setting where the agent accumulates knowledge from many tasks, with limited interactions within each task.

In this work, we consider a *meta-reinforcement-learning* problem made of a sequence of **related tasks**. We leverage this structural task similarity to obtain model estimators with faster convergence as more tasks are seen. The central question of our work is:

Can we meta-learn the model across tasks and adapt the effective planning horizon accordingly?

We take inspiration from the *Average Regret-Upper-Bound Analysis* [ARUBA] framework (Khodak et al., 2019) to generalize planning loss bounds to the meta-RL setting. A high-level, intuitive outline of our approach is presented in Fig. 1. **Our main contributions** are as follows:

- We formalize planning in a model-based meta-RL setting as an *average planning loss* minimization problem, and we propose an algorithm to solve it.
- Under a structural *task-similarity* assumption, we prove a novel high-probability task-averaged regret upper-bound on the planning loss of our algorithm, inspired by ARUBA. We also demonstrate a way to learn the task-similarity parameter σ on-the-fly. To the best of our knowledge, this is a first formal (ARUBA-style) analysis to show that meta-RL can be more efficient than RL.
- Our theoretical result highlights a new dependence of the planning horizon on the size of the within-task data m *and* on the number of tasks T. This observation allows us to propose two heuristics to adapt the planning horizon given the overall sample-size.

2 Preliminaries

Reinforcement Learning. We consider tabular Markov Decision Processes (MDPs) $\mathcal{M} = \langle \mathcal{S}, \mathcal{A}, R, P, \gamma_{\texttt{eval}} \rangle$, where $\mathcal{S}$ is a finite set of states, $\mathcal{A}$ is a finite set of actions and we denote the set cardinalities as $S = |\mathcal{S}|$ and $A = |\mathcal{A}|$. For each state $s \in \mathcal{S}$, and for each available action $a \in \mathcal{A}$, the probability vector $P(\cdot \mid s, a)$ defines a transition model over the state space and is a probability distribution in a set of feasible models $\mathcal{D}_P \subset \Delta_S$, where Δ_S the probability simplex of dimension $S - 1$. We denote $\Sigma \leq 1$ the diameter of $\mathcal{D}_P$. A policy is a function $\pi : \mathcal{S} \to \mathcal{A}$ and it characterizes the agent's behavior.

We consider the bounded reward setting, *i.e.*, $R \in [0, R_{\max}]$ and without loss of generality we set $R_{\max} = 1$ (unless stated otherwise). Given an MDP, or task, M, for any policy π, let $V^{\pi}_{M,\gamma} \in \mathbb{R}^S$ be the value function when evaluated in MDP M with discount factor $\gamma \in (0, 1)$ (potentially different from $\gamma_{\texttt{eval}}$); defined as $V^{\pi}_{M,\gamma}(s) = \mathbb{E}\sum_{t=0}^{\infty}(\gamma^t R_{s_t} \mid s_0 = s)$. The goal of the agent is to find an optimal policy, $\pi^{\star}_{M,\gamma} = \arg\max_{\pi} \mathbf{E}_{s\sim\rho} V^{\pi}_{M,\gamma}(s)$ where $\rho > 0$ is any positive measure, denoted $\pi^{\star}$ when there is no ambiguity. For given state and action spaces and reward function $(\mathcal{S}, \mathcal{A}, R)$, we denote Π_{γ} the set of *potentially* optimal policies for discount factor γ: $\Pi_{\gamma} = \{\pi \mid \exists\, P \text{ s.t. } \pi = \pi^{\star}_{M,\gamma} \text{ where } M = \langle \mathcal{S}, \mathcal{A}, R, P, \gamma \rangle\}$. We use Big-O notation, $O(\cdot)$ and $\tilde{O}(\cdot)$, to hide respectively universal constants and poly-logarithmic terms in T, S, A and $\delta > 0$ (the confidence level).

Model-based Reinforcement Learning. In practice, the true model of the world is unknown and must be estimated from data. One approach to approximately solve the optimization problem above is to construct a model, $\langle \hat{R}, \hat{P} \rangle$ from data, then find $\pi^{\star}_{\hat{M},\gamma}$ for the corresponding MDP $\hat{M} = \langle \mathcal{S}, \mathcal{A}, \hat{R}, \hat{P}, \gamma \rangle$. This approach is called *model-based RL* or *certainty-equivalence (CE) control*.

Planning with inaccurate models. In this setting, Jiang et al. (2015) define the planning loss as the gap in expected return in MDP M when using $\gamma \leq \gamma_{\texttt{eval}}$ and the optimal policy for an approximate model $\hat{M}$:

$$\mathcal{L}(\hat{M}, \gamma \mid M, \gamma_{\texttt{eval}}) = \| V^{\pi^{\star}_{M,\gamma_{\texttt{eval}}}}_{M,\gamma_{\texttt{eval}}} - V^{\pi^{\star}_{\hat{M},\gamma}}_{M,\gamma_{\texttt{eval}}} \|_{\infty}.$$

Thus, the **optimal effective planning horizon** $(1-\gamma^{\star})^{-1}$ is defined using the discount factor that minimizes the planning loss, *i.e.*, $\gamma^{\star} := \min_{0 \leq \gamma \leq \gamma_{\texttt{eval}}} \mathcal{L}(\hat{M}, \gamma \mid M, \gamma_{\texttt{eval}})$.

Theorem 1. *(Jiang et al. (2015)) Let M be an MDP with non-negative bounded rewards and evaluation discount factor γ_{eval}. Let $\hat{M}$ be the approximate MDP comprising the true reward function of M and the approximate transition model $\hat{P}$, estimated from $m > 0$ samples for each state-action pair. Then, with probability at least $1 - \delta$,*

$$\left\| V^{\pi^{\star}_{M,\gamma_{eval}}}_{M,\gamma_{eval}} - V^{\pi^{\star}_{\hat{M},\gamma}}_{M,\gamma_{eval}} \right\|_{\infty} \leq \frac{\gamma_{eval} - \gamma}{(1-\gamma_{eval})(1-\gamma)} + \frac{2\gamma R_{\max}}{(1-\gamma)^2} \left(\sqrt{\frac{\Sigma}{2m} \log \frac{2SA|\Pi_{\gamma}|}{\delta}} \right) \tag{1}$$

where Σ is upper-bounded by 1 as $P, \hat{P} \in \Delta_S$.

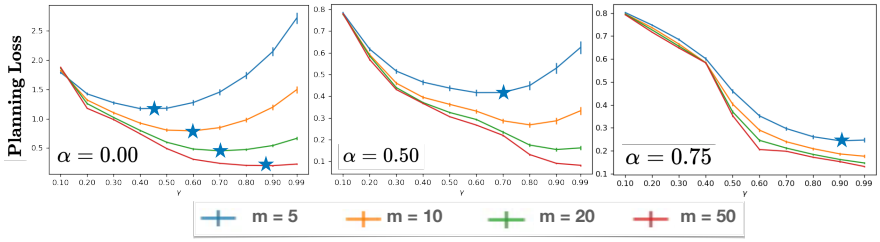


Figure 2: **On the role of incorporating a ground truth prior of transition model on planning horizon.** The planning loss is a function of the discount factor γ and is impacted by incorporating prior knowledge. The learner has $m = 5, 10, 20, 50$ samples per task to estimate the model, corresponding to the curves in each sub figure. Inspecting any of the sub figures, we observe that larger values of m lead to lower planning loss and a larger effective discount factor. Besides, inspecting one value of m across tasks (e.g., $m = 5$), we see that the same effect (lower planning loss and larger effective discount) occurs when the learner puts more weight on the ground truth prior through α.

This result holds for a count-based model estimator (i.e, empirical average of observed transitions) given by a generator model for each pair (s, a). It gives an upper-bound on the planning loss as a function of the guidance discount factor $\gamma < 1$. The result decomposes the loss into two terms: the constant bias which decreases as γ tends to $\gamma_{\texttt{eval}}$, and the variance (or uncertainty) term which increases with γ but decreases as $1/\sqrt{m}$. As $m \to \infty$ that second factor vanishes, but in the low-sample regime the optimal effective planning horizon should trade-off both terms.

Illustration. These effects are illustrated in Fig. 2 on a simple 10-state, 2-action random MDP. The leftmost plot uses the simple count-based model estimator and reproduces the results from Jiang et al. (2015). We then incorporate the true prior (mean model P^o as in Fig 1 and defined above Eq. 3 in Sec. 3.1) in the estimator with a growing mixing factor $\alpha \in (0,1)$: $\hat{P}(m) = \alpha P^o + (1-\alpha)\frac{\sum^i X^i}{m}$. We observe that increasing the weight $\alpha \in (0,1)$ on good prior knowledge enables longer planning horizons and lower planning loss.

Online Meta-Learning and Regret. We consider an online meta-RL problem where an agent is presented with a sequence of tasks $M_1, M_2, ..., M_T$, where for each $t \in [T]$, $M_t = \langle \mathcal{S}, \mathcal{A}, P^t, R, \gamma_{\texttt{eval}} \rangle$, that is, the MDPs only differ from each other by the transition matrix (dynamics model) P^t. The learner must sequentially estimate the model $\hat{P}^t$ for each task t from a batch of m transitions simulated for each state-action pair[1].

Its goal is to minimize the average planning loss also expressed in the form of task averaged regret suffered in planning and defined as

$$\bar{\mathcal{L}}(\hat{M}_{1:T}, \gamma | M_{1:T}, \gamma_{\texttt{eval}}) = \frac{1}{T}\sum_{t=1}^{T} \mathcal{L}(\hat{M}_t, \gamma | M_t, \gamma_{\texttt{eval}}) = \frac{1}{T}\sum_{t=1}^{T} \|V^{\pi^\star_{M_t,\gamma_{\texttt{eval}}}}_{M_t,\gamma_{\texttt{eval}}} - V^{\pi^\star_{\hat{M}_t,\gamma}}_{M_t,\gamma_{\texttt{eval}}}\|_\infty \tag{2}$$

Note that the reference MDP for each term is the true M_t, and the discount factor γ is the same in all tasks. One can see this objective as a stochastic dynamic regret: at each task $t \in [T]$, the learner competes against the optimal policy for the *current* true model, as opposed to competing against the best fixed policy in hindsight used in classical definitions of regret.

Note that **our dynamic regret is different from the one considered in ARUBA** (Khodak et al., 2019). They consider the fully online setting where the data is observed as an arbitrary stream within each task, and each comparator is simply the minimum of the within-task loss in hindsight. In our model, however, we assume we have access to a simulator which allows us to get i.i.d transition samples as a batch at the beginning of each task, and consequently to define our regret with respect to the true generating parameter. One key consequence of this difference is that their regret bounds cannot be directly applied to our setting, and we prove new results further below.

3 Planning with Online Meta-Reinforcement Learning

We here formalize planning in a model-based meta-RL setting. We start by specifying our structural assumption in Sec. 3.1, present our approach and explain the proposed algorithms `POMRL` and `ada-POMRL` in Sec. 3.2. Our main result is a high-probability upper bound on the average planning loss under the assumed structure, presented as Theorem 2.

3.1 Structural Assumption Across Tasks

In many practical scenarios, the key reason to employ meta-learning is for the learner to leverage a **task-similarity** (or task variance) structure across tasks. Bounded task similarity is becoming a core assumption in the analysis of recent meta learning (Khodak et al., 2019) and multi-task (Cesa-Bianchi et al., 2021) online learning algorithms. In this work, we exploit the structural assumption that for all $t \in [T]$, $P^t \sim \mathcal{P}$ centered at some fixed but unknown $P^o \in \Delta_S^{S\times A}$ and such that for any (s,a),

$$\|P^t_{s,a} - P^o_{s,a}\|_\infty \leq \sigma = \max_{(s,a)} \sigma(s,a) \quad \text{a.s.} \tag{3}$$

This also implies that $\max_{t,t'} \|P^t_{s,a} - P^{t'}_{s,a}\|_\infty \leq 2\sigma$, and that the meta-distribution $\mathcal{P}$ is bounded within a small subset of the simplex. It is immediate to extend our results under a high-probability assumption instead of the almost sure statement above. In our experiments, we will use Gaussian or Dirichlet priors over the simplex, whose moments are bounded with high-probability, not almost surely. Importantly, we will say that a multi-task environment is *strongly structured* when $\sigma < \Sigma$, *i.e.* when the effective diameter of the models is smaller than that of the entire feasible space.

[1] So a total of mSA samples.

3.2 Our Approach

For simplicity we shall assume throughout that the rewards are known[2] and focus on learning and planning with an approximate dynamics model. We assume that for each task $t \in [T]$ we have access to a simulator of transitions (Kearns et al., 2002) providing m i.i.d. samples $(X^{t,i}_{s,a})_{i=1..m} \in \mathcal{S}^m \sim P^t(\cdot|s,a)$ (categorical distribution). For each (s,a), we can compute an empirical estimator for each $s' \in [S]$: $\bar{P}^t_{s,a}(s') = \sum_{i=1}^m \mathbb{1}\{X^{t,i}_{s,a} = s'\}/m$, with naturally $\sum_{s'} \bar{P}^t_{s,a}(s') = 1$. We perform meta-RL via alternating minimizing a batch *within-task* regularized least-squares loss, and an outer-loop step where we optimize the regularization to optimally balance bias and variance of the next estimator.

Estimating dynamics model via regularized least squares. We adapt the standard technique of meta-learned regularizer (see e.g. Baxter (2000); Cella et al. (2020) for supervised learning and bandit respectively) to this model estimation problem. At each round, the current model is estimated by minimizing a regularized least square loss: for a given regularizer h_t (to be specified below)[3] and parameter $\lambda_t > 0$ for each $(s,a) \in \mathcal{S} \times \mathcal{A}$ we solve

$$\hat{P}^t_{(s,a)} = \underset{P_{(s,a)} \in \Delta_S}{\arg\min} \left(\ell_t(P_{(s,a)}|X^{1:m,t}, h_t, \lambda_t) = \|\frac{1}{m}\sum_{i=1}^m \mathbb{1}\{X^{t,i}_{s,a}\} - P_{(s,a)}\|_2^2 + \lambda_t \|P_{(s,a)} - h_t\|_2^2 \right), \tag{4}$$

where we use $\mathbb{1}\{X^{t,i}_{s,a}\}$ to denote the one-hot encoding of the state into a vector in $\mathbb{R}^S$. Importantly, h_t and λ_t are meta-learned in the outer-loop (see below) and affect the bias and variance of the resulting estimator since they define the within-task loss. The solution of equation 4 can be computed in closed form as a convex combination of the empirical average (count-based) and the prior: $\hat{P}^t = \alpha_t h_t + (1-\alpha_t)\bar{P}^t$ where $\alpha_t = \frac{\lambda_t}{1+\lambda_t}$ is the current mixing parameter.

Outer-loop: Meta-learning the regularization. At the beginning of task $1 < t \leq T$, the learner has already observed $t-1$ *related but different* tasks. We define h_t as an **average of Means (AoM)**:

$$h^t_{(s,a)} \leftarrow \hat{P}^{o,t}_{(s,a)} = \frac{1}{t-1}\sum_{j=1}^{t-1} \frac{\sum_{i=1}^m \mathbb{1}\{X^{j,i}_{(s,a)}\}}{m} := \frac{1}{t-1}\sum_{j=1}^{t-1} \bar{P}^j_{(s,a)}. \tag{5}$$

Deriving the mixing rate. To set α_t, we compute the Mean Squared Error (MSE) of $\hat{P}^t_{(s,a)}$, and minimize an upper bound (see details in Appendix B): $\mathrm{MSE}(\hat{P}^t_{(s,a)}) \leq \alpha_t^2\sigma^2(1+\frac{1}{t}) + (1-\alpha_t)^2\frac{1}{m}$, which leads to $\alpha_t = \frac{1}{\sigma^2(1+1/t)m+1}$.

Algorithm 1 depicts the complete pseudo code. We note here that `POMRL` (σ) assumes, for now, that the underlying task-similarity parameter σ is known, and we discuss a fully empirical extension further below (See Sec. 4). The learner does not know the number of tasks a priori and tasks are faced sequentially online. The learner performs meta-RL alternating between within-task estimation of the dynamics model $\hat{P}^t$ via a batch of m samples for that task, and an outer loop step to meta-update the regularizer $\hat{P}^{o,t+1}$ alongside the mixing rate α_{t+1}. For each task, we use a γ-`Selection-Procedure` to choose planning horizon $\gamma^* \leq \gamma_{\texttt{eval}}$. We defer the details of this step to Sec. 6 as it is non-trivial and only a partial consequence of our theoretical analysis. Next, the learner performs planning with an imperfect model $\hat{P}^t$. For planning, we use dynamic

[2]We note that additionally estimating the reward is a straightforward extension of our analysis and would not change the implications of our main result.

[3]In principle, this loss is well defined for any regularizer h_t but we specify a meta-learned one and prove that it induces good performance.

programming, in particular policy iteration (a combination of policy evaluation, and improvement), and value iteration to obtain the optimal policy $\pi^{\star}_{\hat{P}^t,\gamma^*}$ for the corresponding MDP $\hat{M}_t$.

Algorithm 1: `POMRL` (σ) – Planning with Online Meta-Reinforcement Learning

Input: Set meta-initialization $\hat{P}^{o,1}$ to uniform, task-similarity $(\sigma(s,a))$ a matrix of size $S \times A$, mixing rate $\alpha_1 = 0$, and $\gamma_{\texttt{eval}}$

for *task* $t \in [T]$ **do**

  for t^{th} *batch of* m *samples* **do**

    $\hat{P}^t(m) = (1-\alpha_t)\frac{1}{m}\sum_{i=1}^{m} X_i + \alpha_t \hat{P}^{o,t}$ `// regularized least squares minimizer.`

    $\gamma^{\star} \leftarrow \gamma$-`Selection-Procedure`$(m, \alpha_t, \sigma, T, S, A)$

    $\pi^{\star}_{\hat{P}^t,\gamma^*} \leftarrow$ `Planning`$(\hat{P}^t(m))$ `//`

    Output: $\pi^{\star}_{\hat{P}^t,\gamma^*}$

  Update $\hat{P}^{o,t+1}$, $\alpha_{t+1} = \frac{1}{\sigma^2(1+1/t)m+1}$ `// meta-update AoM (Eq. 5) and mixing rate`

3.3 Average Regret Bound for Planning with Online-meta-learning

Our main theoretical result below controls the average regret of `POMRL` (σ), a version of Alg. 1 with additional knowledge of the underlying task structure, *i.e.*, the true $\sigma > 0$.

Theorem 2. *Using the notation of Theorem 1, we bound the average planning loss equation 2 for* `POMRL` (σ)*:*

$$\bar{\mathcal{L}} \leq \frac{\gamma_{eval} - \gamma}{(1-\gamma_{eval})(1-\gamma)} + \frac{2\gamma S}{(1-\gamma)^2}\tilde{O}\left(\frac{\sigma + \sqrt{\frac{1}{T}\left(\sigma + \sqrt{\sigma^2 + \frac{\Sigma}{m}}\right)}}{\sigma^2 m + 1} + \frac{\sigma^2 m \sqrt{\frac{\Sigma}{m}}}{\sigma^2 m + 1}\right) \tag{6}$$

with probability at least $1-\delta$*, where* $\sigma^2 < 1$ *is the measure of the task-similarity and* $\sigma = \max_{(s,a)} \sigma(s,a)$*.*

The proof of this result is provided in Appendix D and relies on a new concentration bound for the meta-learned model estimator. The last term on the r.h.s. corresponds to the uncertainty on the dynamics. First we verify that if $T = 1$ and m grows large, the second term dominates and is equivalent to $\tilde{O}(\sqrt{\frac{\Sigma}{m}})$ (as $\sigma^2/(\sigma^2 m + 1) \to 0$), which is similar to that of Jiang et al. (2015) as there is no meta-learning, with an additional $O(\frac{1}{m})$ but second order term due to the introduced bias. Then, if m is fixed and small, for small enough values of σ^2 (typically $\sigma < 1/\sqrt{m}$), the first term dominates and the r.h.s. boils down to $\tilde{O}\left((\sigma + \frac{1}{\sqrt{m}})/\sqrt{T}\right)$. This highlights the interplay of our structural assumption parameter σ and the amount of data m available at each round. The regimes of the bound for various similarity levels are explored empirically in Sec. 5 (Q3). We also show the dependence of the regret upper bound on m and T for a fixed σ, in Appendix Fig. F3.

Implications for degree of task-similarity *i.e.*, σ **values.** Our bound suggests that the degree of improvement you can get from meta learning scales with the task similarity σ instead of the set size Σ. Thus, for $\sigma \leq \Sigma$, performing meta learning with Algorithm 1 guarantees better learning measured via our improved regret bound when there is underlying structure in the problem space which we formalize through Eq. 3. Should σ be large, the techniques will still hold and our bounds will simply scale accordingly.

When $\sigma = 0$, all tasks are exactly the same. Indeed, the mixing rate $\alpha_t \approx 1$ for all t, so our algorithm boils down to returning the average of means $\hat{P}^{o,t}$ for each task, which simply corresponds to solving the tasks as a continuous, uninterrupted stream of batches from the nearly same model that $\hat{P}^{o,t}$ aggregates. Unsurprisingly, our bound recovers that of (Jiang et al., 2015, Theorem 1): the bound below reflects that we have to estimate only one model in a space of "size" Σ with mT samples.

$$\bar{\mathcal{L}} \leq \frac{\gamma_{\texttt{eval}} - \gamma}{(1-\gamma_{\texttt{eval}})(1-\gamma)} + \frac{2\gamma S}{(1-\gamma)^2}\tilde{O}\left(\sqrt{\frac{\Sigma}{mT}}\right) \tag{7}$$

When $\sigma = 1$, then $\sigma = \Sigma = 1$, then the meta-learning assumption is not relevant but our bound remains valid and gracefully degrades to reflect it. We need to estimate T models each with m samples. Then the second term $\frac{1}{\sqrt{m}}$ reflects the usual estimation error for each task while the first term is an added bias (second order in $\frac{1}{m}$) due to our regularization to our mean prior P^o that is not relevant here.

$$\bar{\mathcal{L}} \leq \frac{\gamma_{\texttt{eval}} - \gamma}{(1-\gamma_{\texttt{eval}})(1-\gamma)} + \frac{2\gamma S}{(1-\gamma)^2}\tilde{O}\Big(\frac{1}{m}\Big(1 + \frac{1}{\sqrt{T}}(1 + \sqrt{1 + \frac{1}{m}})\Big) + \frac{1}{\sqrt{m}}\Big) \tag{8}$$

Connections to ARUBA. As explained earlier, our metric is not directly comparable to that of ARUBA (Khodak et al., 2019) but it is interesting to make a parallel with the high-probability average regret bounds proved in their Theorem 5.1. They also obtain an upper bound in $\tilde{O}(1/\sqrt{m} + 1/\sqrt{mT})$ if one upper bounds their average within-task regret $\bar{U} \leq B\sqrt{m}$.

Remark 1 (Role of the task similarity σ in Eq. 2)**.** *When $\sigma > 0$, `POMRL` naturally integrates each new data batch into the model estimation. The knowledge of σ is necessary to obtain this exact and intuitive update rule, and our theory only covers `POMRL` equipped with this prior knowledge, but we discuss how to learn and plug-in $\hat{\sigma}_t$ in practice. Note that it would be possible to extend our result to allow for using the empirical variance estimator with tools like the Bernstein inequality, but we believe this it out of the scope of this work as it would essentially give a similar bound as obtained in Theorem 2 with an additional lower order term in $O(1/T)$, and it would not provide much further intuition on the meta-planning problem we study.*

4 Practical Considerations: Adaption On-The-Fly

In this section we propose a variant of `POMRL` that meta learns the task similarity parameter, which we call `ada-POMRL` . We compare the two algorithms empirically in a 10 state, 2 action MDP with closely related underlying task structure across $T = 15$ tasks (details of the experiment setup are deferred to Sec. 5).

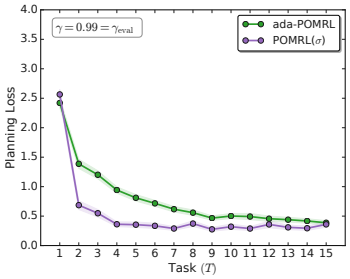


Figure 3: `ada-POMRL` enables meta-learning the task-similarity on-the-fly with a performance gap for the initial set of tasks as compared to the oracle `POMRL` , but improves with more tasks

Performance of `POMRL` . Recall that `POMRL` is primarily learning the regularizer and assumes the knowledge of the underlying task similarity (i.e. σ). We observe in Fig. 3 that with each round $t \in T$ `POMRL` is able to plan better as it learns and adapts the regularizer to the incoming tasks. The convergence rate and final performance corroborates with our theory.

Can we also meta-learn the task-similarity parameter? In practice, the parameter σ may not be known and must be estimated online and plugged in (see Appendix C for details). Alg. 2 `ada-POMRL` uses Welford's algorithm to compute an online estimate of the variance after every task using the model estimators, and simply plugs-in this estimate wherever `POMRL` was using the true value. From the perspective of `ada-POMRL` , `POMRL` is an "oracle", i.e. the underlying task-similarity is known. However, in most practical scenarios, the learner does not have this information a priori. We compare empirically `POMRL` and `ada-POMRL` on a strongly structured problem ($\sigma \approx 0.01$) in Fig. 3 and observe that meta-learning the underlying task structure allows `ada-POMRL` to adapt to the incoming tasks accordingly. Adaptation on-the-fly with `ada-POMRL` comes comes at a cost *i.e.*, the performance gap in comparison to `POMRL` but eventually converges albeit with a slower rate. This is intuitive and a similar theoretical guarantee applies (See Remark 1).

This online estimation of σ means that `ada-POMRL` now requires an initial value for $\hat{\sigma}_1$, which is a choice left to the practitioner, but will only affect the results of a finite number of tasks at the beginning. Using $\hat{\sigma}_1$ too small will gives a slightly increased weight to the prior in initial tasks, which is not desirable as the latter is not yet learned and will result in an increased bias. On the other hand, setting $\hat{\sigma}_1$ too large (i.e close to 1/2) will decrease the weight of the prior and increase the variance of the returned solution; in particular, in cases where the true σ is small, a large initialization will slow down convergence and we observe empirical larger gaps between `POMRL` and `ada-POMRL` . In the extreme case where $\sigma \approx 0$, a large initialization will drastically

slow down `ada-POMRL` as it will take many tasks before it *discovers* that the optimal behavior is essentially to aggregate the batches.

Algorithm 2: `ada-POMRL` – Planning with Online Meta-Reinforcement Learning

Input: Set meta-initialization $\hat{P}^{o,1}$ to uniform, initialize $(\hat{\sigma})_1$ as a matrix of size $S \times A$, mixing rate $\alpha_1 = 0$, and $\gamma_{\texttt{eval}}$

for *task* $t \in [T]$ **do**

  for t^{th} *batch of* m *samples* **do**

    $\hat{P}^t(m) = (1-\alpha_t)\frac{1}{m}\sum_{i=1}^{m} X_i + \alpha_t \hat{P}^{o,t}$ // `regularized least squares minimizer.`

    $\gamma^\star \leftarrow \gamma$-`Selection-Procedure`$(m, \alpha_t, \sigma_t, T, S, A)$

    $\pi^\star_{\hat{P}^t,\gamma} \leftarrow$ `Planning`$(\hat{P}^t(m))$ // $\forall \gamma \leq \gamma_{\texttt{eval}}$

    Output: $\pi^\star_{\hat{P}^t,\gamma}$

  Update $\hat{P}^{o,t+1}$, $\hat{\sigma}_{t+1} \leftarrow$ `Welford's online algorithm`$\left((\hat{\sigma_o})_t, \hat{P}^{o,t+1}, \hat{P}^{o,t}\right)$ // `meta-update AoM (Eq. 5) and task-similarity parameter.`

  Update $\alpha_{t+1} = \frac{1}{\hat{\sigma}_{t+1}{}^2(1+1/t)m+1}$ // `meta-update mixing rate, plug` $\max(\sigma_{S\times A})$

Tasks vary only in certain states and actions. Thus far, we considered a *uniform* notion of task similarity as Eq. 3 holds for any (s, a). However, in many practical settings the transition distribution might remains the same for most part of the state space but only vary on some states across different tasks. These scenarios are hard to analyse in general because local changes in the model parameters do not always imply changes in the optimal value function nor necessarily modify the optimal policy. Our Theorem 2 still remains valid, but it may not be tight when the meta-distribution has non-uniform noise levels. More precisely our concentration result of Theorem 1 in Appendix D remains locally valid for each (s, a) pair and one could easily replace the uniform σ with local $\sigma_{(s,a)}$, but this cannot directly imply a stronger bound on the average planning loss. Indeed, in our experiments, in both `POMRL` and `ada-POMRL` , the parameter σ and $\hat{\sigma}$ respectively, are $S \times A$ matrices of state-action dependent variances resulting in state-action dependent mixing rate α_t.

5 Experiments

We now study the empirical behavior of planning with online meta-learning in order to answer the following questions: **Q1.**Does meta-learning a good initialization of dynamics model facilitate improved planning accuracy for the choice of $\gamma = \gamma_{\texttt{eval}}$? (Sec. 5.1) **Q2.**Does meta-learning a good initialization of dynamics model enables longer planning horizons? (Sec. 5.2) **Q3.**How does performance depend on the amount of shared structure across tasks *i.e.*, σ? (Sec. 5.3) Source code is provided in the supplementary material.

Setting: For each experiment, we fix a mean model $P^o \in \Delta_S^{S\times A}$ (see below how), and for each new task $t \in [T]$, we sample P^t from a Dirichlet distribution[4] centered at P^o. As prescribed by theory (see Sec.3.2), we set[5] $\sigma \approx 0.01 \lesssim 1/S\sqrt{m}$ unless otherwise specified (see Q3). Note that σ and $\hat{\sigma}$ respectively, are $S \times A$ matrices of state-action dependent variances that capture the directional variance as we used Dirichlet distributions as priors and these have non-uniform variance levels in the simplex, depending on how close to the simplex boundary the mean is located. Aligned with our theory, we use the max of the σ matrices resulting in the aforementioned single scalar value. As in Jiang et al. (2015), P^o (and each P^t) characterizes a random chain MDP with $S = 10$ states[6] and $A = 2$ actions, which is drawn such that, for each state–action pair, the transition function $P(s, a, s')$ is constructed by choosing randomly $k = 5$ states whose probability is set to 0. Then we draw the value of the $S - k$ remaining states uniformly in $[0, 1]$ and we normalize the resulting vector.

[4]The variance of this distribution is controlled by its coefficient parameters $\alpha_{1:S}$: the larger they are, the smaller is the variance. More details on our choices are given in Appendix F.1. Dirichlet distributions with small variance satisfy the high-probability version of our structural assumption 3 for $\sigma = \max_i \sigma_i$

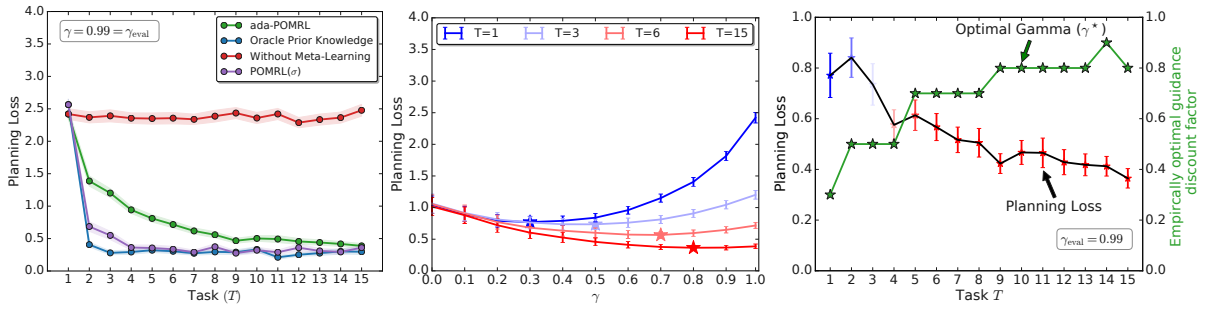
[5]Our priors are multivariate Dirichlet distribution in dimension S so we divide the theoretical rate by S to ensure the max bounded by $1/\sqrt{m}$. See App. F for implementation details.

[6]We provide additional experiments with varying size of the state space in Appendix Fig. F5.

5.1 Meta-reinforcement learning leads to improved planning accuracy for [$\gamma_{\texttt{eval}}$]. [Q1.]

We consider the aforementioned problem setting with a total of $T = 15$ closely related tasks and focus on the planning loss gains due to improved model accuracy. We fix $\gamma = \gamma_{\texttt{eval}}$, a rather naive γ-`Selection-Procedure` and show the planning loss of `POMRL` (Alg. 1) with the following **baselines**: 1) **Oracle Prior Knowledge** knows a priori the underlying task structure (P^o, σ) and uses estimator (Eq. 4) with exact regularizer P^o and optimal mixing rate $\alpha_t = \frac{1}{\sigma^2(1+1/t)m+1}$, 2) **Without Meta-Learning** simply uses $\hat{P}^t = \bar{P}^t$, the count-based estimated model using the m samples seen in each task, 3) `POMRL` (Alg. 1) meta-learns the regularizer but knows apriori the underlying task structure, and 4) `ada-POMRL` (Alg. 2) meta-learns not only the regularizer, but also the underlying task-similarity online. The oracle is a strong baseline that provides a minimally inaccurate model and should play the role of an "empirical lower bound". For all baselines, the number of samples per task $m = 5$. Results are averaged over 100 independent runs. Besides, we also propose and empirically validate competitive heuristics for γ-`Selection-Procedure` in Sec. 6. Besides, we also run another baseline called Aggregating($\alpha = 1$), that simply ignores the meta-RL structure and just plans assuming there is a single task (See Appendix F.2).

Inspecting Fig. 4(a), we can see that our approach `ada-POMRL` (green) results in decreasing per-task planning loss as more tasks are seen, and decreasing variance as the estimated model gets more stable and approaches the optimal value returned by the oracle prior knowledge baseline (blue). On the contrary, without meta-learning (red), the agent struggles to cope as it faces new tasks every round, and its performance does not improve. `ada-POMRL` gradually improves as more tasks are seen whilst adaptation to learned task-similarity on-the-fly which is the primary cause of the performance gap in `ada-POMRL` and `POMRL` . Importantly, no prior knowledge about the underlying task structure enables a more practical algorithm with the same theoretical guarantees (See Sec. 4). Recall that oracle prior knowledge is a strong baseline as it corresponds to both known task structure and regularizer.



(a) (b) (c)

Figure 4: **Planning with Online Meta-Learning. (a) Per-task planning loss** of our algorithms `POMRL` and `ada-POMRL` compared to an Oracle, and Without Meta-learning baselines. All methods use a fixed $\gamma = \gamma_{\texttt{eval}} = 0.99$. **(b) `ada-POMRL` 's planning loss** decreases as more tasks are seen. Markers denote the γ that minimizes the planning loss in respective tasks. Error bars show standard error. **(c) `ada-POMRL` 's empirically optimal guidance discount factor** (right y axis) depicts the effective planning horizon, *i.e.*, one that minimizes the planning loss. Optimal γ aka the effective planning horizon is larger with online meta-learning. Planning loss (left y axis) shows the minimum planning loss achieved by the agent in that round T. Results are averaged over 100 independent runs and error bars represent 1-standard deviation.

5.2 Meta-learning the underlying task structure enables longer planning horizons. [Q2.]

We run `ada-POMRL` for $T = 15$ (with $\sigma \approx 0.01$) as above and report planning losses for a range of values of guidance γ factors. Results are averaged over 100 independent runs and displayed on Fig. 4(b). We observe in Fig. 4(b) when the agent has seen fewer tasks T, an intermediate value of the discount is optimal, *i.e.*, one that minimizes the task-averaged planning loss ($\gamma^\star < 0.5$). In the presence of strong underlying structure across tasks, **as the agent sees more tasks, the effective planning horizon** ($\gamma^\star > 0.7$) **shifts to a larger value** - one that is closer to the gamma used for evaluation ($\gamma_{\texttt{eval}} = 0.99$).

As we incorporate the knowledge of the underlying task distribution, *i.e.*, meta-learned initialization of the dynamics model, we note that the adaptive mixing rate α_t puts increasing amounts of weight on the shared task-knowledge. Note that this conforms to the effect of increasing weight on the model initialization that we observed in Fig. 2. As predicted by theory, the per-task planning loss decreases as T grows and is minimized for progressively larger values of γ, meaning for longer planning horizons (See Fig. 4(c)). In addition, Appendix Fig. F4 depicts the effective planning horizon individually for `ada-POMRL` , Oracle and without meta learning baselines.

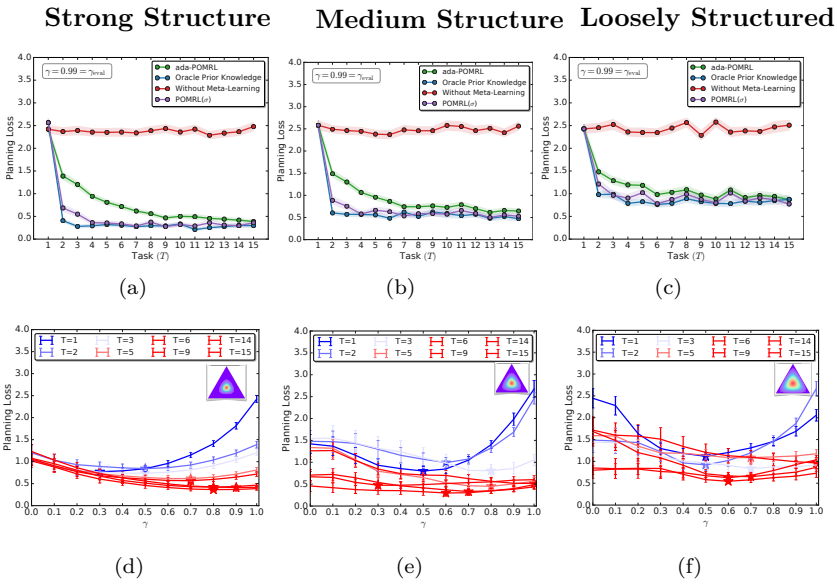


Figure 5: **`POMRL` and `ada-POMRL` are robust to varying task-similarity** σ for a small fixed amount of data $m = 5$ available at each round $t \in T$. A small value of σ reflects the fact that tasks are closely related to each other and share a good amount of structure whereas a much larger value indicates loosely related tasks (simplex plots illustrate the meta-distribution in dimension 2). In the former case, meta-learning the shared structure alongside a good model initialization leads to most gains. In the latter, the learner struggles to cope with new unseen tasks which differ significantly. Error bars represent 1-standard deviation of uncertainty across 100 independent runs.

5.3 `POMRL` and `ada-POMRL` perform consistently well for varying task-similarity. [Q3.]

We have thus far studied scenarios where the learner can exploit strong task structure, *i.e.*, $\sigma \approx 0.01 < 1/(S\sqrt{m})$ (for low data per task *i.e.*, $m = 5$) is small and we now illustrate the other regimes discussed in Section 3.2. We show that our algorithms remain consistently good for all amounts of task-similarity.

We let σ vary to cover the **three regimes**: $\sigma \approx 0.01$ corresponding to fast convergence, $\sigma = 0.025$ is in the intermediate regime (needs longer T), and $\sigma = 0.047$ is the loosely structured case where we don't expect much meta-learning to help improve model accuracy. The small inset figures in Fig. 5 represent the task distribution in the simplex. In all cases, `ada-POMRL` estimates σ online and we report the planning losses for a range of γ's. Inspecting Fig. 5, we observe that while in the presence of closely related tasks (Fig. 5(a)) all methods perform well (except without meta-learning). As the underlying task structure decreases (for intermediate regime in Fig. 5(b)), both `POMRL` and `ada-POMRL` remain consistent in their performance as compared to the Oracle Prior Knowledge baseline. When the underlying tasks are loosely structured (as in Fig. 5(c)), `ada-POMRL` and `POMRL` can still perform well in comparison to other baselines.

Next, we report and discuss the planning loss plot for `ada-POMRL` for the three cases are shown in Figures 5(d), 5(e), and 5(f) respectively. An intermediate value of task-similarity (Fig. 5(e)) still leads to gains, albeit at a lower speed of convergence. In contrast, a large value of $\sigma = 0.047$ indicates little structure across tasks resulting in minimal gains from meta-learning here as seen in Fig. 5(f). The learner struggles to learn a good initialization of the model dynamics as there is no natural one. All planning loss curves remain U-shaped and overall higher with an intermediate optimal guidance γ value (0.5). However, `ada-POMRL` does not do worse

overall than the initial run $T = 1$, meaning that while there is not a significant improvement, our method does not hurt performance in loosely structured tasks[7]. Recall that `ada-POMRL` has no apriori knowledge of the number of tasks (T), or the underlying task-structure (σ) *i.e.*, adaptation is on-the-fly.

6 Adaptation of Planning Horizon γ

We now propose and empirically validate two heuristics to design an adaptive schedule for γ based on existing work (Sec. 6.1) and on our average regret upper bound (Sec. 6.2).

6.1 Schedule adapted from Dong et al. (2021) [$\gamma = f(m, \alpha_t, \sigma_t, T)$]

Dong et al. (2021) study a continuous, never-ending RL setting. They divide the time into growing phases $(T_t)_{t\geq 0}$, and tune a discount factor $\gamma_t = 1 - 1/T_t^{1/5}$. We adapt their schedule to our problem, where the time is already naturally divided into tasks: for each $t \geq 0$, we define the phase size T_t and the corresponding γ_t as

$$T_0 = m, \quad T_t = \frac{SA}{L}\Big(\underbrace{(1-\alpha_t)m + \alpha_t m(t-1)}_{\text{efficient sample size}}\Big), \quad \gamma_t = 1 - \frac{1}{T_t^{1/5}},$$

where L is the maximum trajectory length. The size of each T_t, $t \geq 1$, is controlled by an "efficient sample size" which includes a combination of the current task's samples and of the samples observed so far, as used to construct our estimator in `POMRL` .

6.2 Using the upper bound to guide the schedule [$\gamma = \min\{1, \gamma_0 + \tilde{\gamma}\}$]

Having a second look at Theorem 2, we see that the r.h.s. is a function of γ of the form

$$U : \gamma \mapsto \frac{1}{1-\gamma_{\texttt{eval}}} + \frac{1}{\gamma - 1} + C_{m,T,S,A,\sigma,\delta}\frac{\gamma}{(1-\gamma)^2},$$

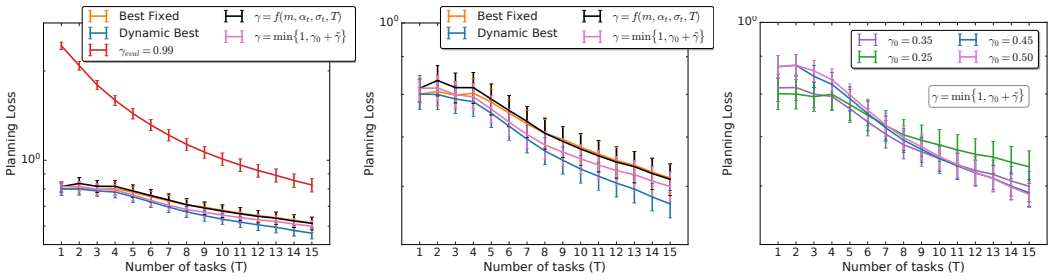
where the first term is positive and monotonically decreasing on $(0, \gamma_{\texttt{eval}})$ and the second term is positive and monotonically increasing on $(0, 1)$. We simplify and scale this constant, keeping only problem-related terms: $C_t = (\frac{1}{\sqrt{t}}(\sigma + \frac{1}{\sqrt{m}})/(\sigma^2 m + 1) + \sigma^2 m \frac{1}{\sqrt{m}}/(\sigma^2 m + 1)$, which is of the order of the constant in equation 6. Optimizing γ by using the function U with constant C does not lead to a principled analytical value strictly speaking because U is derived from an upper bound that may be loose and may not reflect the true shape of the loss w.r.t. γ, but we may use the resulting growth schedule to guide our choices online. In general, the existence of a strict minimum for U in $(0, 1)$ is not always guaranteed: depending on the values of $C \approx C_{m,T,S,A,\sigma}$, the function may be monotonic and the minimum may be on the edges. We give explicit ranges in the proposition below, proved in Appendix E.

Proposition 1. *The existence of a strict minimum in* $(0, 1)$ *is determined by* $C = C_{m,T,S,A,\sigma,\delta}$ *(which can be computed) as follows:*

$$\tilde{\gamma} = \begin{cases} 0 & \text{if } C \geq 1 \\ 1 & \text{if } C < 1/2 \\ \frac{1-C}{1+C} & \text{otherwise, i.e if } 1/2 < C < 1 \end{cases}$$

We use these values as a guide. Typically, when $T = 1$ and m is small, the multiplicative term C is large and the bound is not really informative (concentration has not happened yet), and γ should be small, potentially close to but not equal to zero. As a heuristic, we propose to simply offset $\tilde{\gamma}$ by an additional γ_0 such that the guidance discount factor is $\gamma = \min\{1, \gamma_0 + \tilde{\gamma}\}$, where γ_0 should be reasonably chosen by the practitioner to allow for some short-horizon planning at the beginning of the interaction. Empirically, $\gamma_0 =\in (0.25, 0.50)$ seems reasonable for our random MDP setting as it corresponds to the empirical minima on Fig 4(b).

[7]The theoretical bound may lead to think that the average planning loss is higher due to the introduced bias, but in practice we do not observe that, which means our bound is pessimistic on the second order terms.



(a) (b) (c)

Figure 6: **Adapting the planning horizon during online meta-learning reduces planning loss. (a)** Planning with online-meta learning shows that *all* baselines outperform using a constant discount factor. **(b)** Zoomed in plot of average planning loss over the progression of tasks T shows competitive performance with the proposed schedule of $\gamma = f(m, \alpha_t, \sigma_t, T)$ beating best-fixed as more tasks are seen. The γ schedule $\gamma = \min\{1, \gamma_0 + \tilde{\gamma}\}$ using the upper bound as a guidance beats the best-fixed and is very competitive to the dynamic-best baseline. **(c)** Using the upper bound to guide the schedule significantly outperforms $\gamma_{\texttt{eval}}$ and is shown for $\gamma_0 \in (0.25, 0.50)$. Error bars depict 1-standard error for 600 independent runs.

6.3 Empirical Validation

Next, we empirically test the proposed schedules for adaptation of discount factors. We consider the setup described in Sec. 5 with 15 tasks in a 10-state, 2-action random MDP distribution of tasks with $\sigma \approx 0.01$. In Fig. 6, we plot the planning loss obtained by `POMRL` with our schedules, a fixed $\gamma_{\texttt{eval}}$ and two strong baselines: *best fixed* which considers the best fixed value of discount over all tasks estimated in hindsight and *dynamic best* which considers the best choice if we had used the optimal $\gamma^\star$ in each round as in Fig. 4(c). It is important to note that *dynamic best* is a lower bound that we cannot outperform.

We observe in Fig. 6(a) that $\gamma_{\texttt{eval}}$ results in a very high loss, potentially corresponding to trying to plan too far ahead despite model uncertainty. Upon inspecting Fig. 6(b), we observe that the proposed $\gamma = f(m, \alpha_t, \sigma_t, T)$ obtains similar performance to *best fixed* and is within the significance range of the lower bound. Our second heuristic, $\gamma = \min\{1, \gamma_0 + \tilde{\gamma}\}$ obtains similarly good performance, as seen in Fig. 6(b). Fig. 6(c) shows the effect of different values of γ_0 in the prescribed range. These results provide evidence that it is possible to adapt the planning horizon as a function of the problem's structure (meta-learned task-similarity) and sample sizes. Adapting the planning horizon online is an open problem and beyond the scope of our work.

7 Discussion and Future Work

We presented connections between planning with inaccurate models and online meta-learning via a high-probability task-averaged regret upper-bound on the planning loss that primarily depends on task-similarity σ as opposed to the entire search space Σ. Algorithmically, we demonstrate that the agent can use its experience in each task *and* across tasks to estimate both the transition model and the distribution over tasks. Meta-learning the underlying structure across tasks and a good initialization of transition model across tasks enables longer planning horizons.

Fully online meta-learning: A limitation of our work is that we assume access to a simulator which allows us to get i.i.d. transition samples as a batch at the beginning of each task, as opposed to a fully online setting. One idea in this direction is to replace the current *average of means* regularizer by an online meta-gradient step closer to the ARUBA and MAML. It is unclear how to obtain similar concentration results as Theorem 2 with such gradient updates. **Non-stationary meta-distribution**: We considered that the underlying task-distribution is stationary. Many real-world scenarios have (slow or sudden) drifts in the underlying distribution itself, e.g. weather. A promising future direction is to consider non-stationary environments where the optimal initialization varies over time. **Guarantees beyond the tabular case:** a fruitful direction is to derive similar guarantees amenable for transition models with function approximation. While the work of Khodak et al. (2019) makes strides for supervised learning, it is non-trivial to extend these bounds for RL theory.

References

Ron Amit, Ron Meir, and Kamil Ciosek. Discount factor as a regularizer in reinforcement learning. In *International conference on machine learning*, pp. 269–278. PMLR, 2020.

Dilip Arumugam, David Abel, Kavosh Asadi, Nakul Gopalan, Christopher Grimm, Jun Ki Lee, Lucas Lehnert, and Michael L Littman. Mitigating planner overfitting in model-based reinforcement learning. *arXiv preprint arXiv:1812.01129*, 2018.

Maria-Florina Balcan, Mikhail Khodak, and Ameet Talwalkar. Provable guarantees for gradient-based meta-learning. In *International Conference on Machine Learning*, pp. 424–433. PMLR, 2019.

Jonathan Baxter. A model of inductive bias learning. *Journal of artificial intelligence research*, 12:149–198, 2000.

Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling. The arcade learning environment: An evaluation platform for general agents. *Journal of Artificial Intelligence Research*, 47:253–279, 2013.

David Blackwell. Discrete dynamic programming. *The Annals of Mathematical Statistics*, pp. 719–726, 1962.

Rich Caruana. Multitask learning. *Machine learning*, 28(1):41–75, 1997.

Leonardo Cella, Alessandro Lazaric, and Massimiliano Pontil. Meta-learning with stochastic linear bandits. In *International Conference on Machine Learning*, pp. 1360–1370. PMLR, 2020.

Nicolò Cesa-Bianchi, Pierre Laforgue, Andrea Paudice, and Massimiliano Pontil. Multitask online mirror descent. *arXiv preprint arXiv:2106.02393*, 2021.

Giulia Denevi, Carlo Ciliberto, Dimitris Stamos, and Massimiliano Pontil. Learning to learn around a common mean. *Advances in Neural Information Processing Systems*, 31, 2018.

Giulia Denevi, Dimitris Stamos, Carlo Ciliberto, and Massimiliano Pontil. Online-within-online meta-learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Shi Dong, Benjamin Van Roy, and Zhengyuan Zhou. Simple agent, complex environment: Efficient reinforcement learning with agent state. *arXiv preprint arXiv:2102.05261*, 2021.

William Fedus, Carles Gelada, Yoshua Bengio, Marc G Bellemare, and Hugo Larochelle. Hyperbolic discounting and learning over multiple horizons. *arXiv preprint arXiv:1902.06865*, 2019.

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

Chelsea Finn, Aravind Rajeswaran, Sham Kakade, and Sergey Levine. Online meta-learning. In *International Conference on Machine Learning*, pp. 1920–1930. PMLR, 2019.

Sebastian Flennerhag, Yannick Schroecker, Tom Zahavy, Hado van Hasselt, David Silver, and Satinder Singh. Bootstrapped meta-learning. *arXiv preprint arXiv:2109.04504*, 2021.

Sebastian Flennerhag, Tom Zahavy, Brendan O'Donoghue, Hado van Hasselt, András György, and Satinder Singh. Optimistic meta-gradients. In *Sixth Workshop on Meta-Learning at the Conference on Neural Information Processing Systems*, 2022.

Vincent François-Lavet, Guillaume Rabusseau, Joelle Pineau, Damien Ernst, and Raphael Fonteneau. On overfitting and asymptotic bias in batch reinforcement learning with partial observability. *Journal of Artificial Intelligence Research*, 65:1–30, 2019.

Nan Jiang, Alex Kulesza, Satinder Singh, and Richard Lewis. The dependence of effective planning horizon on model accuracy. In *Proceedings of the 2015 International Conference on Autonomous Agents and Multiagent Systems*, pp. 1181–1189. International Foundation for Autonomous Agents and Multiagent Systems, 2015.

Michael Kearns, Yishay Mansour, and Andrew Y Ng. A sparse sampling algorithm for near-optimal planning in large markov decision processes. *Machine learning*, 49(2):193–208, 2002.

Mikhail Khodak, Maria-Florina Balcan, and Ameet Talwalkar. Adaptive gradient-based meta-learning methods. *arXiv preprint arXiv:1906.02717*, 2019.

Levente Kocsis and Csaba Szepesvári. Bandit based monte-carlo planning. In *European conference on machine learning*, pp. 282–293. Springer, 2006.

Jelena Luketina, Sebastian Flennerhag, Yannick Schroecker, David Abel, Tom Zahavy, and Satinder Singh. Meta-gradients in non-stationary environments. In *ICLR Workshop on Agent Learning in Open-Endedness*, 2022.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529, 2015.

Junhyuk Oh, Matteo Hessel, Wojciech M Czarnecki, Zhongwen Xu, Hado van Hasselt, Satinder Singh, and David Silver. Discovering reinforcement learning algorithms. *arXiv preprint arXiv:2007.08794*, 2020.

Marek Petrik and Bruno Scherrer. Biasing approximate dynamic programming with a lower discount factor. *Advances in neural information processing systems*, 21, 2008.

Joelle Pineau. The machine learning reproducibility checklist. *arxiv*, 2019.

Juergen Schmidhuber and Rudolf Huber. Learning to generate artificial fovea trajectories for target detection. *International Journal of Neural Systems*, 2(01n02):125–134, 1991.

Terence Tao and Van Vu. Random matrices: universality of local spectral statistics of non-hermitian matrices. *The Annals of Probability*, 43(2):782–874, 2015.

Sebastian Thrun and Lorien Pratt. Learning to learn: Introduction and overview. In *Learning to learn*, pp. 3–17. Springer, 1998.

Harm Van Seijen, Hado Van Hasselt, Shimon Whiteson, and Marco Wiering. A theoretical and empirical analysis of expected sarsa. In *2009 ieee symposium on adaptive dynamic programming and reinforcement learning*, pp. 177–184. IEEE, 2009.

Zhongwen Xu, Hado van Hasselt, and David Silver. Meta-gradient reinforcement learning. *arXiv preprint arXiv:1805.09801*, 2018.

Tom Zahavy, Zhongwen Xu, Vivek Veeriah, Matteo Hessel, Junhyuk Oh, Hado P van Hasselt, David Silver, and Satinder Singh. A self-tuning actor-critic algorithm. *Advances in neural information processing systems*, 33, 2020.