Offline Meta Reinforcement Learning with In-Distribution Online Adaptation

Jianhao Wang [*1] **Jin Zhang** [*1] **Haozhe Jiang** [1] **Junyu Zhang** [2] **Liwei Wang** [3] **Chongjie Zhang** [1]

Abstract

Recent offline meta-reinforcement learning (meta-RL) methods typically utilize task-dependent behavior policies (e.g., training RL agents on each individual task) to collect a multi-task dataset. However, these methods always require extra information for fast adaptation, such as offline context for testing tasks. To address this problem, we first formally characterize a unique challenge in offline meta-RL: transition-reward distribution shift between offline datasets and online adaptation. Our theory finds that out-of-distribution adaptation episodes may lead to unreliable policy evaluation and that online adaptation with in-distribution episodes can ensure adaptation performance guarantee. Based on these theoretical insights, we propose a novel adaptation framework, called In-Distribution online Adaptation with uncertainty Quantification (IDAQ), which generates in-distribution context using a given uncertainty quantification and performs effective task belief inference to address new tasks. We find a return-based uncertainty quantification for IDAQ that performs effectively. Experiments show that IDAQ achieves state-of-the-art performance on the Meta-World ML1 benchmark compared to baselines with/without offline adaptation.

1. Introduction

Human intelligence is capable of learning a wide variety of skills from past experiences and can adapt to new environments by transferring skills with limited experience. Current reinforcement learning (RL) has surpassed human-level performance (Mnih et al., 2015; Silver et al., 2017; Hafner et al., 2019). However, in many real-world applications, RL encounters two major challenges: multi-task efficiency and costly online interactions. In multi-task settings, such as robotic manipulation or locomotion (Yu et al., 2020b), agents are expected to solve new tasks in few-shot adaptation using previously learned knowledge. Moreover, collecting sufficient exploratory interactions is usually expensive or dangerous in robotics (Rafailov et al., 2021), autonomous driving (Yu et al., 2018), and healthcare (Gottesman et al., 2019). One popular paradigm for breaking this practical barrier is *offline meta reinforcement learning* (offline meta-RL; Li et al., 2020b; Mitchell et al., 2021), which trains a meta-RL agent with pre-collected offline multi-task datasets and enables fast policy adaptation to unseen tasks.

Recent offline meta-RL methods have been proposed to utilize a multi-task dataset collected by task-dependent behavior policies (Li et al., 2020b; Dorfman et al., 2021). They show promise by solving new tasks with few-shot adaptation. However, existing offline meta-RL approaches require additional information or assumptions for fast adaptation. For example, FOCAL (Li et al., 2020b) and MACAW (Mitchell et al., 2021) use offline contexts for meta-testing tasks. BOReL (Dorfman et al., 2021) and SMAC (Pong et al., 2022) employ few-shot online adaptation, in which the former assumes known reward functions, and the latter assumes unsupervised online samples (without rewards) are available in offline meta-training. Therefore, achieving effective online fast adaptation without extra information remains an open problem for offline meta-RL.

To approach meta-testing relying on online experience in offline meta-RL, we first characterize a unique conundrum: ***transition-reward distribution shift*** between offline datasets and online adaptation, complementary to *state-action distribution shift* in offline RL (Levine et al., 2020). As illustrated in Figure 1, we propose a motivating example to visualize the transition-reward distribution shift. In this example, the robot aims to choose the correct path to reach the *diamond* (💎) in three tasks. During task-dependent data collection, the offline multi-task dataset consists of all *successful episodes* (⟶) through expert behavior policies. After offline meta-training on the given dataset, the robot needs to fast adapt to a (unknown) meta-testing task, i.e., task 3 shown in Figure 1. In the first adaptation episode, the robot does not know the identification of meta-testing task. It may try the middle path, stop in front of the stone,

[*]Equal contribution [1]Institute for Interdisciplinary Information Sciences, Tsinghua University [2]Huazhong University of Science and Technology [3]Institute of Artificial Intelligence, Peking University. Correspondence to: Chongjie Zhang <zhangchongjie@gmail.com>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

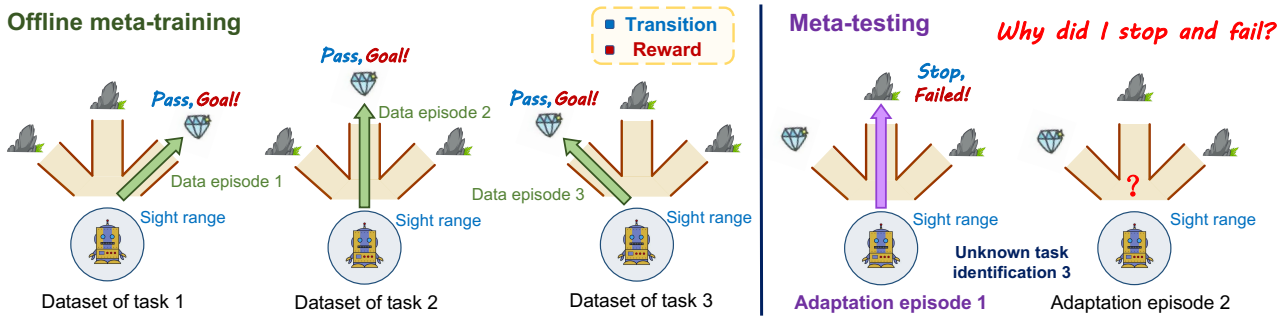


Figure 1. Illustration of transition-reward distribution shift between offline training and online adaptation.

and fail. The reward and transition of this failed *adaptation episode* (→) is out-of-distribution from the offline dataset because the trajectories of the given dataset are successful. This out-of-distribution context will *confuse* the agent in inferring task belief since it is not encountered during offline meta-training. To formalize this phenomenon, we build a theory from the perspective of Bayesian RL (BRL; Duff, 2002; Zintgraf et al., 2019), which maintains a task belief given the context history and learns a meta-policy on the belief states. Our theory finds that (i) the transition-reward distribution shift exists and may lead to unreliable policy evaluation, (ii) filtering out out-of-distribution episodes in online adaptation can ensure the performance guarantee, and (iii) meta-policies with Thompson sampling (Strens, 2000) can generate in-distribution episodes.

The transition-reward distribution shift induces the ***inconsistency dilemma*** of experience between offline meta-training and online meta-testing. We can choose either to trust the *offline dataset* (→) or to trust *new experience* (→) and continue online exploration. The latter may not be able to collect sufficient data in few-shot adaptation to learn a good policy only on online data. Therefore, we adopt the former strategy and, inspired by our theory, propose a novel context-based online adaptation framework, called ***I**n-**D**istribution online **A**daptation with uncertainty **Q**uantification* (IDAQ). To align online experience with the offline dataset, IDAQ distinguishes in-distribution context using a given uncertainty quantification, performs task belief updating, and samples "task hypotheses" to solve new tasks. We investigate three uncertainty quantifications to measure the confidence that adaptation episodes are in-distribution, and find that IDAQ with a greedy return-based quantification can perform effectively in complex domains. To serve intuitions in Figure 1, IDAQ will continue to sample other "task hypotheses" (i.e., try other paths) during meta-testing and infer the unknown task 3 using in-distribution adaptation episode (left).

Our main contribution is to formalize a specific challenge (i.e., transition-reward distribution shift), reveal theoretical insights for offline meta-RL with online adaptation, and furthermore propose a novel in-distribution online adaptation framework with theoretical motivation. To our best knowledge, our method is the first to conduct in-distribution online fast adaptation in offline meta-RL. We extensively evaluate the performance of IDAQ in didactic problems proposed by prior work (Rakelly et al., 2019; Zhang et al., 2021) and Meta-World ML1 benchmark with 50 tasks (Yu et al., 2020b). Empirical results show that IDAQ significantly outperforms baselines with fast online adaptation, and achieves better or comparable performance than offline adaptation baselines with expert context.

2. Notations and Preliminaries

We defer the detailed background to Appendix A.1.

2.1. Standard Meta-RL

The standard meta-RL (Finn et al., 2017; Rakelly et al., 2019) deals with a distribution $p(\kappa)$ over *Markov Decision Processes* (MDPs), where each task $\kappa_i \sim p(\kappa)$ presents a finite-horizon MDP (Zintgraf et al., 2019), which is defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, H, P^{\kappa_i}, R^{\kappa_i})$, including state space $\mathcal{S}$, action space $\mathcal{A}$, reward space $\mathcal{R}$, planning horizon H, transition function $P^{\kappa_i}(s'|s,a)$, and reward function $R^{\kappa_i}(r|s,a)$. In this paper, we assume dynamics function P and reward function R may vary across tasks and share a common structure. The meta-RL algorithms repeatedly sample batches of tasks to train a meta-policy. In meta-testing, the agent aims to rapidly adapt a good policy for new tasks drawn from $p(\kappa)$ within N adaptation episodes.

From a perspective of Bayesian RL (BRL; Ghavamzadeh et al., 2015), recent meta-RL methods (Zintgraf et al., 2019) utilize a *Bayes-adaptive MDP* (BAMDP; Duff, 2002) to formalize standard meta-RL. A BAMDP is defined as a tuple $M^+ = \left(\mathcal{S}^+, \mathcal{A}, \mathcal{R}, H^+, P_0^+, P^+, R^+\right)$, where $\mathcal{S}^+ = \mathcal{S} \times \mathcal{B}$ is hyper-state space, $\mathcal{B}$ is task belief space, a task belief b is the posterior over MDPs given the previous experience, $H^+ = N \times H$ is planning horizon, $P_0^+\left(s_0^+\right)$ is initial hyper-state distribution, $P^+\left(s_{t+1}^+ \middle| s_t^+, a_t, r_t\right)$ is transition function, and $R^+\left(r_t \middle| s_t^+, a_t\right)$ is reward function. The objective of meta-RL agents is to find a meta-policy $\pi^+\left(a_t \middle| s_t^+\right)$ to maximize the *online policy evaluation* $\mathcal{J}_{M^+}\left(\pi^+\right)$.

2.2. Offline Meta-RL

In the offline meta-RL setting (Li et al., 2020b), a meta-learner only has access to an offline multi-task dataset $\mathcal{D}^+$ and is not allowed to interact with the environment during meta-training. Recent offline meta-RL methods (Dorfman et al., 2021) always utilize task-dependent behavior policies $p(\mu|\kappa)$, which represents the random variable of the behavior policy $\mu(a|s)$ conditioned on the random variable of the task κ. For brevity, we overload $[\mu] = p(\mu|\kappa)$. Similar to offline RL (Yin & Wang, 2021), we assume that $\mathcal{D}^+$ consists of multiple i.i.d. trajectories that are collected by executing task-dependent policies $[\mu]$ in M^+. Denote the reward and transition distribution of the task-dependent offline data collection (Jin et al., 2021) by $\mathbb{P}_{M^+,[\mu]}\left(r_t, s_{t+1} \middle| s_t^+, a_t\right)$.

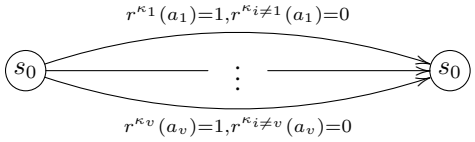
During meta-training, offline RL (Liu et al., 2020; Chen & Jiang, 2019) approximates *offline policy evaluation* for a batch-constrained policy π^+ by sampling from an offline dataset $\mathcal{D}^+$, which is denoted by $\mathcal{J}_{\mathcal{D}^+}\left(\pi^+\right)$ and called *Approximate Dynamic Programming* (ADP; Bertsekas & Tsitsiklis, 1995). Note that a batch-constrained policy π^+ only selects actions within the dataset $\mathcal{D}^+$ to avoid *extrapolation error* (Fujimoto et al., 2019). In meta-testing, RL agents perform online fast adaptation using a meta-trained policy π^+ in a new task $\kappa_{test} \sim p(\kappa)$. The reward and transition distribution of online data collection in M^+ (Zintgraf et al., 2019) is denoted by $\mathbb{P}_{M^+,\pi^+}\left(r_t, s_{t+1} \middle| s_t^+, a_t\right)$.

2.3. Offline Meta-Training with Task Embedding

In this paper, we follow the algorithmic framework of *Task Embeddings for Actor-Critic RL* (PEARL; Rakelly et al., 2019). The task identification κ is modeled by a latent task embedding z, called "task hypothesis". The offline meta-training learns a context encoder $q(z|\boldsymbol{c})$, a policy $\pi(a|s,z)$, and a value function $Q(s,a,z)$ from a given dataset, where $\boldsymbol{c}$ is the context information including states, actions, rewards, and next states. The encoder $q(z|\boldsymbol{c})$ infers a task belief about the latent task variable z based on the received context. Denote the prior distribution with $\boldsymbol{c} = \emptyset$ by $q(z)$. To distinguish different task identifications from an offline dataset, recent offline meta-RL (Li et al., 2020b; Yuan & Lu, 2022) apply the contrastive loss on the representation of latent task embedding z. The policy π and value function Q are trained with RL losses on the given z.

3. Theory: Transition-Reward Distribution Shift in Offline Meta-RL

Recently, offline meta-RL (Dorfman et al., 2021) faces a new challenge: *transition-reward distribution shift* between offline datasets and online adaptation. We first formalize this data distribution mismatch from the perspective of Bayesian RL (BRL; Zintgraf et al., 2019) and prove its existence. Our theory shows that the transition-reward distribution shift may lead to unreliable policy evaluation and that in-distribution online adaptation can provide consistent performance guarantee. In addition, we prove that meta-policies with Thompson sampling (Strens, 2000) can generate in-distribution online adaptation episodes.



$$p(\kappa_i) = \tfrac{1}{v}, p(\mu_i|\kappa_i) = 1, \text{and } \mu_i(a_i|s_0) = 1$$

Figure 2. A concrete example, which has v meta-RL tasks, one state, v actions, v behavior policies, horizon $H = 1$ in an episode, and v adaptation episodes, where $v \geq 3$.

3.1. Transition-Reward Distribution Shift

We define the distributional shift as follows.

Definition 1 (Transition-Reward Distribution Shift)**.** *In a BAMDP M^+, for each task-dependent behavior policy $[\mu]$ and batch-constrained meta-policy π^+, the transition-reward distribution shift is defined by that there exists a pair of $\left(s_t^+, a_t\right)$ with executing π^+ in M^+, s.t.,*

$$\mathbb{P}_{M^+,[\mu]}\left(r_t, s_{t+1} \middle| s_t^+, a_t\right) \neq \mathbb{P}_{M^+,\pi^+}\left(r_t, s_{t+1} \middle| s_t^+, a_t\right), \tag{1}$$

where $\mathbb{P}_{M^+,[\mu]}, \mathbb{P}_{M^+,\pi^+}$ are the reward and transition distribution of offline data collection by $[\mu]$ and online data collection by π^+, respectively, whose formal definition are deferred to Appendix A.1.4.

This definition utilizes the discrepancy between offline and online data collection to characterize the joint distribution gap of reward and transition. Note that in offline data collection $\mathbb{P}_{M^+,[\mu]}$, the behavior policies $p(\mu|\kappa)$ can vary based on task identification, whereas the online data collection $\mathbb{P}_{M^+,\pi^+}$ is the expected reward and transition distribution across the task distribution $p(\kappa)$.

Theorem 1. *There exists a BAMDP M^+ with task-dependent behavior policies $[\mu]$ such that, for any batch-constrained meta-policy π^+, the transition-reward distribution shift between $\mathbb{P}_{M^+,[\mu]}$ and $\mathbb{P}_{M^+,\pi^+}$ occurs.*

To prove the existence of distributional shift, we construct an offline meta-RL setting shown in Figure 2, which has v meta-RL tasks $\{\kappa_1, \ldots, \kappa_v\}$, v behavior policies $\{\mu_1, \ldots, \mu_v\}$, and $v \geq 3$. The task distribution $p(\kappa)$ is uniform, the behavior policy of task κ_i is μ_i, and each behavior policy μ_i will perform a_i. After data collection, RL agents will offline meta-train policies on a given dataset $\mathcal{D}^+$ and fast adapt to a meta-testing task $\kappa_{test} \sim p(\kappa)$ within v online episodes.

In this example, for any action a_i in s_0^+, the offline data collection $\mathbb{P}_{M^+,[\mu]}\left(r = 1 \middle| s_0^+, a_i\right) = 1$ since expert task-dependent behavior policies all collect data with reward 1.

During online meta-testing, for any batch-constrained meta-policy π^+ selects an action a_i in s_0^+, the online data collection $\mathbb{P}_{M^+,\pi^+}\left(r=1 \,\middle|\, s_0^+, a_i\right) = \frac{1}{v}$ because there is the probability of $\frac{1}{v}$ to sample a meta-testing task κ_i, whose reward function of a_i is 1. Thus, $\mathbb{P}_{M^+,[\mu]} \neq \mathbb{P}_{M^+,\pi^+}$.

3.2. Data Distribution Matters for Online Adaptation

To investigate the impact of data distribution mismatch, we analyze the gap of policy evaluation between offline dataset $\mathcal{J}_{\mathcal{D}^+}$ and online adaptation $\mathcal{J}_{M^+}$ in offline meta-RL.

Proposition 1. *There exists a BAMDP M^+ with task-dependent behavior policies such that, for any batch-constrained meta-policy π^+, (i) RL agents will visit out-of-distribution hyper-states and (ii) the gap between offline policy evaluation $\mathcal{J}_{\mathcal{D}^+}(\pi^+)$ and online policy evaluation $\mathcal{J}_{M^+}(\pi^+)$ is at least $\frac{H^+-1}{2}$.*

Proposition 1 states that RL agents will go out of the distribution of the offline dataset $\mathcal{D}^+$ due to the shifts in the reward and transition distribution. Thus, the offline policy evaluation of π^+ in meta-training cannot provide a reference for the online mest-testing. For example in Figure 2, the agent will visit out-of-distribution belief states when receiving reward 0 with probability $1-\frac{1}{v}$ in s_0^+. In addition, the offline policy evaluation $\mathcal{J}_{D^+}(\pi^+) = H^+ = v$ since the dataset D^+ only contains reward 1. For each π^+, we have $\mathcal{J}_{M^+}(\pi^+) \leq \frac{H^++1}{2}$ and detailed proof is deferred to Appendix A.2. Hence, the gap between $\mathcal{J}_{\mathcal{D}^+}(\pi^+)$ and $\mathcal{J}_{M^+}(\pi^+)$ is at least $\frac{H^+-1}{2}$.

To address this inconsistency dilemma, we choose to trust the offline dataset within few-shot online adaptation and derive the following theorem.

Theorem 2. *In a BAMDP M^+, for each task-dependent behavior policy $[\mu]$, denoting a transformed BAMDP $\overline{M}^+$ by incorporating $[\mu]$ into the belief of M^+, we have (i) for feasible Bayesian belief updating, $\overline{M}^+$ confines the agent in the in-distribution hyper-states, (ii) for each $\bar{\pi}^+$, the distribution of $\mathbb{P}_{\overline{M}^+,[\mu]}$ and $\mathbb{P}_{\overline{M}^+,\bar{\pi}^+}$ matches, and (iii) policy evaluation $\mathcal{J}_{\overline{\mathcal{D}}^+}(\bar{\pi}^+)$ and $\mathcal{J}_{\overline{M}^+}(\bar{\pi}^+)$ will be asymptotically consistent, as the offline dataset grows.*

To achieve in-distribution online adaptation, transformed BAMDPs incorporate additional information about offline data collection into the beliefs of BAMDPs. We prove that transformed BAMDPs require RL agents to filter out out-of-distribution episodes to support feasible belief updating of behavior policies. In this way, the distribution of reward and transition between offline and online data collection coincides, which can provide the guarantee of consistent policy evaluation between $\mathcal{J}_{\overline{\mathcal{D}}^+}(\bar{\pi}^+)$ and $\mathcal{J}_{\overline{M}^+}(\bar{\pi}^+)$. Theorem 2 shows that we can meta-train policies with offline policy evaluation and utilize in-distribution online adaptation to guarantee the final performance in meta-testing.

3.3. Generating In-Distribution Online Adaptation

In this subsection, we will incorporate Thompson sampling (Strens, 2000) into the meta-policies to generate in-distribution episodes during online adaptation as follows.

Theorem 3. *In a transformed BAMDP $\overline{M}^+$, for each batch-constrained meta-policy with Thompson sampling $\bar{\pi}^{+,T}$ in a meta-testing task $\kappa_{test} \sim p(\kappa)$, there exists a task hypothesis from the current belief, executing $\bar{\pi}^{+,T}$ in κ_{test} can generate in-distribution online adaptation episodes with high probability, as the offline dataset grows.*

Theorem 3 indicates that for each adaptation episode, we can sample task hypotheses from the current task belief and execute $\bar{\pi}^{+,T}$ to interact with the environment until finding an in-distribution episode. For example in Figure 2, after offline meta-training, a meta-policy with Thompson sampling $\bar{\pi}^{+,T}$ will perform a_i with a task hypothesis of κ_i and expect to receive a reward 1. During online meta-testing, κ_{test} is drawn from $p(\kappa)$ and the agent needs to infer the task identification. To achieve in-distribution online adaptation, $\bar{\pi}^{+,T}$ will try various actions according to diverse task hypotheses until sampling an in-distribution episode with a reward 1. Updating the task belief with the in-distribution episode, RL agents can infer and solve this task.

In contrast, when updating task belief using an out-of-distribution episode with a reward 0, the posterior task belief will be out of the offline dataset $\overline{\mathcal{D}}^+$. Note that offline training paradigm can not well-optimize $\bar{\pi}^{+,T}$ on out-of-distribution states (Fujimoto et al., 2019) and policy $\bar{\pi}^{+,T}$ will fail in this case. Moreover, Thompson sampling is very popular in context-based deep meta-RL (Rakelly et al., 2019) and we will generalize these theoretical implications.

4. IDAQ: In-Distribution Online Adaptation with Uncertainty Quantification

In our setting, offline meta-RL contains two phases: offline meta-training and online adaptation. For offline meta-training, we employ an off-the-shelf context-based algorithm, e.g., FOCAL (Li et al., 2020b), which follows the learning paradigm of latent task embedding (see Section 2.3). In this section, we will focus on investigating a practical scheme to address the major challenge of the transition-reward distribution shift during online adaptation. Motivated by our theory in Section 3, we aim to ***distinguish*** whether an adaptation episode is in the distribution of the offline dataset, and utilize meta-policies with Thompson sampling (Strens, 2000) to generate in-distribution online adaptation. Therefore, we will introduce a novel context-based online adaptation algorithm, called ***I**n-**D**istribution online **A**daptation with uncertainty **Q**uantification* (IDAQ), which infers in-distribution context for solving meta-testing

tasks. The overall algorithm of IDAQ is illustrated in Algorithm 1. IDAQ consists of two main components: (i) a general in-distribution online adaptation framework, and (ii) a plug-in uncertainty quantification function. We will describe these components in detail as follows.

4.1. In-Distribution Online Adaptation Framework

As motivated by Theorem 3, our adaptation protocol adopts the popular framework of *Thompson sampling* (Rakelly et al., 2019) for online meta-testing. IDAQ will iteratively update posterior task belief based on online interactions with environment and execute the meta-policy with a sampled "task hypothesis". For in-distribution online adaptation, IDAQ utilizes a given uncertainty quantification $\mathbb{Q}(\tau)$ to empirically measure the confidence that online experience $\{\tau_i\}$ are in-distribution. Note that the context encoder $q(z|\boldsymbol{c})$ (i.e., a task inference module) is meta-trained in the offline dataset and cannot handle out-of-distribution adaptation (Mendonca et al., 2020), where $\boldsymbol{c}$ is the episode-based context. To realize reliable task belief updating, IDAQ needs to estimate a reference threshold δ and defines the in-distribution context

$$\boldsymbol{c}_{in} = \{\tau_i \,|\mathbb{Q}(\tau_i) \leq \delta, \forall \tau_i \in \boldsymbol{c}\}\,. \tag{2}$$

In this way, IDAQ will perform a two-stage paradigm of online adaptation: (i) a *Reference Stage* to estimate the uncertainty threshold δ and (ii) an *Iterative Updating Stage* to update the in-distribution context $\boldsymbol{c}_{in}$, posterior task belief $q(z|\boldsymbol{c}_{in})$, and execution meta-policy $\pi(a|s,z)$.

Reference Stage collects n_r online adaptation episodes $\{\tau_i\}_{i=1}^{n_r}$ using the prior task distribution $q(z)$ and meta-policy $\pi(a|s,z)$ in a meta-testing task κ_{test}. IDAQ will calculate the in-distribution confidence of adaptation episodes $\{\mathbb{Q}(\tau_i)\}_{i=1}^{n_r}$ and estimate the reference threshold δ that is the bottom $k\%$-quantile of $\{\mathbb{Q}(\tau_i)\}_{i=1}^{n_r}$, where k is a hyperparameter to divide the range of uncertainty of in-distribution episodes. Hence, IDAQ can derive the in-distribution context $\boldsymbol{c}_{in}$ and posterior task belief $q(z|\boldsymbol{c}_{in})$.

Iterative Updating Stage will update the posterior task belief $q(z|\boldsymbol{c}_{in})$ in n_i iterations. In each iteration, IDAQ collects an online adaptation episode τ_j using the current task belief $q(z|\boldsymbol{c}_{in})$ and meta-policy $\pi(a|s,z)$ in κ_{test}. When the uncertainty of this episode $\mathbb{Q}(\tau_j)$ is less than the reference threshold δ, IDAQ will update the in-distribution context, i.e., $\boldsymbol{c}_{in} \leftarrow \boldsymbol{c}_{in} \cup \{\tau_j\}$, and derive the posterior task belief $q(z|\boldsymbol{c}_{in})$. The final policy $\pi_{out}(a|s,z)$ is executed with the total in-distribution context $\boldsymbol{c}_{in}$.

4.2. Uncertainty Quantification

Uncertainty quantification is a popular tool for empirically measuring the confidence that data is in the distribution of offline RL (Yu et al., 2020c) or noisy oracle (Ren et al., 2022). In this subsection, we will analyze three practical uncertainty quantifications to adapt IDAQ to complex domains: *Prediction Error*, *Prediction Variance*, and *Return-based*. The empirical evaluation is deferred to Section 5.1.

Algorithm 1 IDAQ: In-Distribution online Adaptation with uncertainty Quantification

Require: An offline dataset $\mathcal{D}^+$, a meta-testing task κ_{test}, the number of iterations n_i, a context-based offline meta-training algorithm $\mathbb{A}$ (i.e., FOCAL), and an in-distribution uncertainty quantification $\mathbb{Q}$
Offline meta-train a context encoder $q(z|\boldsymbol{c})$ and a meta-policy $\pi(a|s,z)$ using an algorithm $\mathbb{A}$ in a dataset $\mathcal{D}^+$ {***Offline meta-training***}
Perform reference stage of online adaptation and estimate the in-distribution threshold δ using $\mathbb{Q}$ {***Start online meta-testing***}
Derive the in-distribution context $\boldsymbol{c}_{in}$ with Eq. (2) and posterior task belief $q(z|\boldsymbol{c}_{in})$
for $t = 1 \ldots n_i$ **do** {***Iterative updating stage***}
  Collect an online adaptation episode using the posterior task belief q and meta-policy π in κ_{test}
  Update the in-distribution context $\boldsymbol{c}_{in}$ using $\mathbb{Q}, \delta$ and derive the posterior task belief $q(z|\boldsymbol{c}_{in})$
end for
Return: π, $q(z|\boldsymbol{c}_{in})$

To realize the uncertainty quantification of prediction error and prediction variance, we adopt a model-based approach to learn an ensemble of L reward and dynamics models $\{r_{\phi_i}(s,a,z), p_{\psi_i}(s,a,z)\}_{i=1}^{L}$ according to the latent task embedding z. We parameterize them by $\{\phi_i, \psi_i\}_{i=1}^{L}$ and optimize $\{r_{\phi_i}, p_{\psi_i}\}_{i=1}^{L}$ on the offline multi-task dataset $\mathcal{D}^+$ by minimizing the MSE loss function during meta-training. Formal loss function is deferred to Appendix B.

Prediction Error quantifies the model error to estimate the confidence that data is trained during offline meta-training. This metric is also called *"curiosity"*, a popular intrinsic reward in exploration of single-task RL (Pathak et al., 2017), which encourages the agent to visit new areas. In offline meta-RL, we utilize this quantification to filter out out-of-distribution adaptation episodes and denote by

$$\mathbb{Q}_{PE}(\tau_i, z) = \frac{1}{HL}\sum_{t=0}^{H-1}\sum_{i=1}^{L}|r_t - r_{\phi_i}(s_t,a_t,z)| + \|s_{t+1} - p_{\psi_i}(s_t,a_t,z)\|_2\,, \tag{3}$$

where z is the "task hypothesis" of the episode τ_i in IDAQ. $\mathbb{Q}_{PE}$ averages model errors across timesteps and an ensemble. The challenge of $\mathbb{Q}_{PE}$ is that the hyperparameter k for the reference threshold δ will be sensitive for different multi-task datasets since the model error of in-distribution episodes can be various.

Prediction Variance captures the epistemic and aleatoric uncertainty of the true models using a bootstrap ensemble

(Yu et al., 2020c). This metric is popular to measure whether data is in the dataset of offline single-task RL (Kidambi et al., 2020). In offline meta-RL, we denote this quantification by

$$\mathbb{Q}_{PV}(\tau_i, z) = \frac{1}{H}\sum_{t=0}^{H-1} \max_{i,j} \left| r_{\phi_i}(s_t, a_t, z) - r_{\phi_j}(s_t, a_t, z) \right| + \left\| p_{\psi_i}(s_t, a_t, z) - p_{\psi_j}(s_t, a_t, z) \right\|_2, \quad (4)$$

where z is the "task hypothesis" of τ_i. $\mathbb{Q}_{PV}$ averages the ensemble discrepancy across timesteps. However, $\mathbb{Q}_{PV}$ cannot handle cases with higher prediction error and lower prediction variance. For example in Figure 2, the learned reward model will deterministically output 1 for each action, in which the prediction error is $1 - \frac{1}{v}$ with no variance.

Return-based uncertainty quantification is our newly designed metric for offline meta-RL with medium or expert datasets. To address the limitations of prediction error and prediction variance, we utilize a bias of offline RL (Fujimoto et al., 2019) that few-shot out-of-distribution episodes generated by an offline-learned meta-policy π usually have lower returns since offline meta-training can not well-optimize meta-policies on out-of-distribution states. Its contrapositive statement is that executing π with higher returns has a higher probability of being in-distribution and online policy evaluation of π presents a good in-distribution confidence:

$$\mathbb{Q}_{RE}\left(\{\tau_i\}_{i=1}^{n_e}\right) = -\frac{1}{n_e}\sum_{i=1}^{n_e}\sum_{t=0}^{H-1} r_t^i, \quad (5)$$

where n_e is the number of episodes generated by π to approximate the online policy evaluation. With the mild assumption (i.e., the bias of offline RL above), we can prove that return-based uncertainty quantification $\mathbb{Q}_{RE}$ can theoretically derive in-distribution contexts using Theorem 3. The formal analysis is deferred to Appendix A.5. Moreover, in empirical, IDAQ can adopt a conservative (i.e., low) reference threshold δ to achieve in-distribution online adaptation. In this case, IDAQ may neglect some in-distribution episodes with lower returns. We will argue that, in medium or expert datasets, our method can utilize informative episodes with higher returns to perform task inference. It is an interesting and exciting future direction to differentiate in-distribution episodes with lower returns in offline meta-RL with online adaptation.

5. Experiments

In this section, we first evaluate the three uncertainty quantifications mentioned in Section 4.2 and empirically demonstrate that the **Return-based** quantification works the best on various task sets. Then we conduct large-scale experiments on Meta-World ML1(Yu et al., 2020a), a popular meta-RL benchmark that consists of 50 robot arm manipulation task sets. Finally, we perform ablation studies to analyze IDAQ's sensitivity to hyper-parameter settings and *dataset qualities*. Datasets are collected by script policies that solve corresponding tasks. We compare against FOCAL (Li et al., 2020b) and MACAW (Mitchell et al., 2021), as well as their online adaptation variants. We also compare against BOReL (Dorfman et al., 2021) . For a fair comparison, we evaluate a variant of BOReL that does not utilize oracle reward functions, as introduced in the original paper (Dorfman et al., 2021). FOCAL is built upon PEARL (Rakelly et al., 2019) and uses contrastive losses to learn context embeddings, while MACAW is a MAML-based (Finn et al., 2017) algorithm and incorporates AWR (Peng et al., 2019). Both FOCAL and MACAW are originally proposed for the offline adaptation settings (i.e., with expert context). For online adaptation, we use online experience instead of expert contexts, and adopt the adaptation protocol of PEARL and MAML, respectively. Evaluation results are averaged over six random seeds, and variance is measured by 95% bootstrapped confidence interval. Detailed hyper-parameter settings are deferred to Appendix C. A didactic example that empirically demonstrates the distributional shift problem proposed in Section 3 is deferred to Appendix D. An open-source implementation of our algorithm is available online[1].

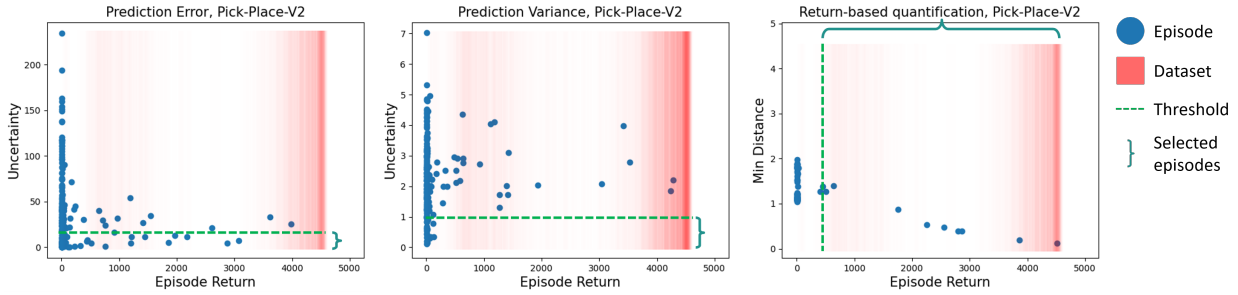
5.1. Evaluation of Uncertainty Quantifications

We evaluate the three uncertainty quantifications mentioned in Section 4.2 on some representative tasks. As shown in Table 1, the **Return-based** quantification significantly outperforms the other two quantifications and the baseline algorithm FOCAL (which uses all online experiences as contexts). To further investigate how these quantifications behave, we illustrate the uncertainty quantifications of various episodes collected in the reference stage on one of the meta-training tasks. As shown in Figure 3(a), the **Prediction Error** quantification cannot find a good reference threshold to distinguish in-distribution episodes. Figure 3(b) shows that the **Prediction Variance** quantification fails and may suffer from situations with higher prediction error and lower prediction variance in the medium or expert datasets. Figure 3(c) illustrates the minimal distance between episodes collected in the reference stage and the offline dataset on one of the meta-training tasks. Results show that episodes with higher returns are closer to the medium or expert datasets, which implies that the **Return-based** quantification can correctly identify in-distribution episodes. Formal distance function and additional visualizations of other tasks are deferred to Appendix E and F.1, respectively.

5.2. Main Results

Following results in Section 5.1, we use the **Return-based** quantification as the default quantification for IDAQ in the following experiments. We evaluate on Meta-World

[1] https://github.com/NagisaZj/IDAQ_Public



(a) (b) (c)

Figure 3. (a) and (b) illustrate the uncertainty quantifications of **Prediction Error** and **Prediction Variance** on episodes collected in the reference stage on one of the meta-training tasks, respectively. Red shades indicate the density of episode returns in the dataset. Both quantifications give low uncertainty measures on the out-of-distribution episodes in the bottom left corner, and fail to identify in-distribution episodes. (c) illustrates the minimal distance between episodes collected in the reference stage and the offline dataset on one of the meta-training tasks. The dotted green line illustrates the reference threshold found by each quantification. Results imply that the **Return-based** quantification successfully identifies in-distribution episodes.

Table 1. Performance of the three uncertainty quantifications and FOCAL on example tasks, a bunch of Meta-World ML1 tasks with normalized scores. "IDAQ+Return" is short for IDAQ with the **Return-based** quantification. For Meta-World tasks, "-V2" is omitted for brevity. "Med" represents results trained on medium quality datasets.

Example Env	IDAQ+Prediction Error	IDAQ+Prediction Variance	IDAQ+Return	FOCAL
Push	0.31 ± 0.13	0.13 ± 0.07	**0.55** ± 0.10	0.34 ± 0.14
Pick-Place	0.07 ± 0.05	0.04 ± 0.03	**0.20** ± 0.03	0.07 ± 0.02
Soccer	0.18 ± 0.03	0.23 ± 0.03	**0.44** ± 0.04	0.11 ± 0.03
Drawer-Close	**1.00** ± 0.00	**0.99** ± 0.01	**0.99** ± 0.02	**0.96** ± 0.04
Reach	**0.87** ± 0.01	0.49 ± 0.03	**0.85** ± 0.03	0.62 ± 0.05
Sweep (Med)	0.15 ± 0.03	0.06 ± 0.02	**0.59** ± 0.13	0.38 ± 0.13
Peg-Insert-Side (Med)	0.03 ± 0.02	0.03 ± 0.01	**0.30** ± 0.14	0.10 ± 0.07
Point-Robot	-5.70 ± 0.05	-21.29 ± 0.85	**-5.10** ± 0.26	-15.38 ± 0.95

ML1(Yu et al., 2020a), a popular meta-RL benchmark that consists of 50 robot arm manipulation task sets. Each task set consists of 50 tasks with different goals. For each task set, we use 40 tasks as meta-training tasks, and remain the other 10 tasks as meta-testing tasks. As shown in Table 2, IDAQ significantly outperforms baselines under the online context setting. With expert contexts, FOCAL and MACAW both achieve reasonable performance. IDAQ achieves better or comparable performance to baselines with expert contexts, which implies that expert contexts may not be necessary for offline meta-RL. Under online contexts, FOCAL fails due to the data distribution mismatch between offline training and online adaptation. MACAW has the ability of online fine-tuning as it is based on MAML, but it also suffers from the distribution mismatch problem, and online fine-tuning can hardly improve its performance within a few adaptation episodes. BOReL fails on most of the tasks, as BOReL without oracle reward functions will also suffer from the distribution mismatch problem, which is consistent with the results in the original paper.

Table 3 shows algorithms' performance on 20 representative Meta-World ML1 task sets, as well a sparse-reward version of Point-Robot and Cheetah-Vel, which are popular meta-RL tasks (Li et al., 2020b). IDAQ achieves remarkable performance in most tasks and may fail in some hard tasks as offline meta-training is difficult. We also find that IDAQ achieves better or comparable performance to baselines with expert contexts on 33 out of the 50 task sets. Detailed algorithm performance on all 50 tasks and comparison to baselines with expert contexts are deferred to Appendix F.2.

5.3. Ablation Studies

We further perform ablation studies on dataset qualities. As shown in Table 1, IDAQ with the **Return-based** quantification achieves state-of-the-art performance on medium-quality datasets. The other two quantifications perform poorly, which may suggest that medium datasets are more challenging to design a good uncertainty quantification and the return-based metric can perform effectively in these settings. Further ablation studies on hyper-parameter settings are deferred to Appendix G. Results demonstrate that IDAQ is generally robust to the choice of hyper-parameters.

Table 2. Algorithms' normalized scores averaged over 50 Meta-World ML1 task sets. Scores are normalized by expert-level policy return.

IDAQ	FOCAL	MACAW	FOCAL with Expert Context	MACAW with Expert Context	BOReL
0.73 ± 0.07	0.53 ± 0.1	0.18 ± 0.1	0.67 ± 0.07	0.68 ± 0.07	0.04 ± 0.01

Table 3. Performance on example tasks, a bunch of Meta-World ML1 tasks with normalized scores.

Example Env	IDAQ	FOCAL	MACAW	BOReL
Coffee-Push	**1.26** ± 0.13	0.66 ± 0.07	0.01 ± 0.01	0.00 ± 0.00
Faucet-Close	**1.12** ± 0.01	1.06 ± 0.02	0.07 ± 0.01	0.13 ± 0.03
Faucet-Open	**1.05** ± 0.02	1.01 ± 0.02	0.08 ± 0.04	0.12 ± 0.05
Door-Close	**0.99** ± 0.00	0.97 ± 0.01	0.00 ± 0.00	0.37 ± 0.19
Drawer-Close	**0.99** ± 0.02	**0.96** ± 0.04	0.53 ± 0.50	0.00 ± 0.00
Door-Lock	**0.97** ± 0.01	0.90 ± 0.02	0.25 ± 0.11	0.14 ± 0.00
Plate-Slide-Back	**0.96** ± 0.02	0.58 ± 0.06	0.21 ± 0.17	0.01 ± 0.00
Dial-Turn	**0.91** ± 0.05	0.84 ± 0.09	0.00 ± 0.00	0.00 ± 0.00
Handle-Press	**0.88** ± 0.05	**0.87** ± 0.02	0.28 ± 0.10	0.01 ± 0.00
Hammer	**0.84** ± 0.06	0.59 ± 0.07	0.10 ± 0.01	0.09 ± 0.01
Button-Press	**0.74** ± 0.08	**0.68** ± 0.14	0.02 ± 0.01	0.01 ± 0.01
Push-Wall	**0.71** ± 0.15	0.43 ± 0.06	0.23 ± 0.18	0.00 ± 0.00
Hand-Insert	**0.63** ± 0.04	0.29 ± 0.07	0.02 ± 0.01	0.00 ± 0.00
Peg-Unplug-Side	**0.56** ± 0.07	0.19 ± 0.09	0.00 ± 0.00	0.00 ± 0.00
Bin-Picking	0.53 ± 0.16	0.31 ± 0.21	**0.66** ± 0.11	0.00 ± 0.00
Soccer	**0.44** ± 0.04	0.11 ± 0.03	**0.38** ± 0.31	0.04 ± 0.02
Coffee-Pull	**0.40** ± 0.05	0.23 ± 0.04	0.19 ± 0.12	0.00 ± 0.00
Pick-Place-Wall	0.28 ± 0.12	0.09 ± 0.04	**0.39** ± 0.25	0.00 ± 0.00
Pick-Out-Of-Hole	0.26 ± 0.25	0.16 ± 0.16	**0.59** ± 0.06	0.00 ± 0.00
Handle-Pull-Side	**0.14** ± 0.04	**0.13** ± 0.09	0.00 ± 0.00	0.00 ± 0.00
Cheetah-Vel	**-171.5** ± 22.00	-287.7 ± 30.6	-234.0 ± 23.5	-301.4 ± 36.8
Point-Robot	**-5.10** ± 0.26	-15.38 ± 0.95	-14.61 ± 0.98	-17.28 ± 1.16
Point-Robot-Sparse	**7.78** ± 0.64	0.83 ± 0.37	0.00 ± 0.00	0.00 ± 0.00

6. Related Work

In the literature, offline meta-RL methods utilize a context-based (Rakelly et al., 2019) or gradient-based (Finn et al., 2017) meta-RL framework to solve new tasks with few-shot adaptation. They utilize the techniques of contrastive learning (Li et al., 2020b; Yuan & Lu, 2022; Li et al., 2020a), more expressive power (Mitchell et al., 2021), or reward relabeling (Dorfman et al., 2021; Pong et al., 2022) with various popular offline single-task RL tricks, i.e., using KL divergence (Wu et al., 2019; Peng et al., 2019; Nair et al., 2020) or explicitly constraining the policy to be close to the dataset (Fujimoto et al., 2019; Zhou et al., 2020). However, these methods always require extra information for fast adaptation, such as offline context for testing tasks (Li et al., 2020b; Mitchell et al., 2021; Yuan & Lu, 2022), oracle reward functions (Dorfman et al., 2021), or available interactions without reward supervision (Pong et al., 2022). To address the challenge, we propose IDAQ, a context-based online adaptation algorithm, to utilize an uncertainty quantification for in-distribution adaptation without requiring additional information.

Similar to single-task offline RL (Levine et al., 2020), SMAC (Pong et al., 2022) finds the policy or state-action distribution shift between learning policies and datasets in offline meta-RL. In this paper, we characterize the transition-reward distribution shift between offline datasets and online adaptation (see Eq. (1)), which is fundamentally different from state-action distribution shift. The reward-transition distribution shift is induced by task-dependent data collection and is unique in offline meta-RL. It specifies the discrepancy of reward and transition distribution given the state-action pairs. When using a behavior meta-policy to collect an offline dataset, reward-transition distribution shift will not appear but state-action or policy distribution shift still exists. SMAC claims that the distribution shift in z-space occurs due to a "policy" mismatch between behavior policies and online adaptation policy. In contrast, the reward-transition distribution shift is a general challenge in

the setting of offline meta-RL. This distribution shift challenge occurs in any offline meta-RL algorithms, including gradient-based algorithms (Finn et al., 2017), and is beyond the z distribution shift tailored for context-based algorithms (Rakelly et al., 2019).

BOReL (Dorfman et al., 2021) focuses on *MDP ambiguity* for task inference. MDP ambiguity and transition-reward distribution shift are two orthogonal challenges in offline meta-RL with task-dependent behavior policies. MDP ambiguity arises from offline datasets with task-dependent data collection, where it may be difficult to differentiate between different MDPs due to narrow sub-datasets of various tasks. On the other hand, the reward-transition distribution shift studies the discrepancy of reward and transition distributions between task-dependent offline dataset and online adaptation. Our work leverages off-the-shelf context-based offline meta-training algorithms, e.g., FOCAL (Li et al., 2020b), for solving the MDP ambiguity problem during offline training, and proposes IDAQ to tackle the reward-transition distribution shift during online adaptation.

PEARL-based online adaptation (Rakelly et al., 2019) may generate out-of-distribution episodes (see Figure 1 and Section 3). Meta-policies with Thompson sampling can generate in-distribution episodes, but the episodes generated by meta-policies with Thompson sampling are not all in-distribution. This is the motivation that we propose our method, which filters out out-of-distribution episodes to support in-distribution online adaptation. Moreover, IDAQ utilizes an exploration method (i.e., Thompson sampling (Strens, 2000)) for in-distribution online adaptation, which is supported by Theorem 3. Thompson sampling is a popular approach for temporally-extended exploration in the literature of meta-RL (Rakelly et al., 2019).

7. Conclusion

This paper formalizes the transition-reward distribution shift in offline meta-RL and introduces IDAQ, a novel in-distribution online adaptation approach. We find that IDAQ with a return-based uncertainty quantification performs effectively in medium or expert datasets. Experiments show that IDAQ can conduct accurate task inference and achieve state-of-the-art performance on Meta-World ML1 benchmark with 50 tasks. IDAQ also performs better or comparably than offline adaptation baselines with expert context, suggesting that offline context may not be necessary for the testing environments. One limitation of the greedy quantification is that it may not utilize in-distribution episodes with lower returns for random datasets and requires more adaptation episodes to sample in-distribution "task hypotheses". Two interesting future directions are to design a more accurate uncertainty quantification and to extend IDAQ to gradient-based in-distribution online adaptation algorithms.

Acknowledgements

The authors would like to thank the anonymous reviewers and Zhizhou Ren for valuable and insightful discussions and helpful suggestions. This work is supported in part by Science and Technology Innovation 2030 - "New Generation Artificial Intelligence" Major Project (No. 2018AAA0100904) and the National Natural Science Foundation of China (62176135).

References

Alekh Agarwal, A. S. Lecture 10: Reinforcement learning. In *COMS E6998.001, Columbia University*. 2017. OpenCourseLecture.

Astrom, K. J. Optimal control of markov decision processes with incomplete state estimation. *J. Math. Anal. Applic.*, 10:174–205, 1965.

Bertsekas, D. P. and Tsitsiklis, J. N. Neuro-dynamic programming: an overview. In *Proceedings of 1995 34th IEEE conference on decision and control*, volume 1, pp. 560–564. IEEE, 1995.

Cassandra, A. R., Kaelbling, L. P., and Littman, M. L. Acting optimally in partially observable stochastic domains. In *Aaai*, volume 94, pp. 1023–1028, 1994.

Chen, J. and Jiang, N. Information-theoretic considerations in batch reinforcement learning. In *International Conference on Machine Learning*, pp. 1042–1051. PMLR, 2019.

Dorfman, R., Shenfeld, I., and Tamar, A. Offline meta reinforcement learning–identifiability challenges and effective data collection strategies. *Advances in Neural Information Processing Systems*, 34, 2021.

Du, S. S., Kakade, S. M., Wang, R., and Yang, L. F. Is a good representation sufficient for sample efficient reinforcement learning? *arXiv preprint arXiv:1910.03016*, 2019.

Duan, Y., Schulman, J., Chen, X., Bartlett, P. L., Sutskever, I., and Abbeel, P. Rl2: Fast reinforcement learning via slow reinforcement learning. *arXiv preprint arXiv:1611.02779*, 2016.

Duff, M. O. *Optimal Learning: Computational procedures for Bayes-adaptive Markov decision processes*. University of Massachusetts Amherst, 2002.

Finn, C., Abbeel, P., and Levine, S. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 1126–1135. JMLR. org, 2017.

Fujimoto, S., Meger, D., and Precup, D. Off-policy deep reinforcement learning without exploration. In *International Conference on Machine Learning*, pp. 2052–2062. PMLR, 2019.

Ghavamzadeh, M., Mannor, S., Pineau, J., Tamar, A., et al. Bayesian reinforcement learning: A survey. *Foundations and Trends® in Machine Learning*, 8(5-6):359–483, 2015.

Gottesman, O., Johansson, F., Komorowski, M., Faisal, A., Sontag, D., Doshi-Velez, F., and Celi, L. A. Guidelines for reinforcement learning in healthcare. *Nature medicine*, 25(1):16–18, 2019.

Hafner, D., Lillicrap, T., Ba, J., and Norouzi, M. Dream to control: Learning behaviors by latent imagination. In *International Conference on Learning Representations*, 2019.

Jin, Y., Yang, Z., and Wang, Z. Is pessimism provably efficient for offline rl? In *International Conference on Machine Learning*, pp. 5084–5096. PMLR, 2021.

Kaelbling, L. P., Littman, M. L., and Cassandra, A. R. Planning and acting in partially observable stochastic domains. *Artificial intelligence*, 101(1-2):99–134, 1998.

Kidambi, R., Rajeswaran, A., Netrapalli, P., and Joachims, T. Morel: Model-based offline reinforcement learning. *Advances in neural information processing systems*, 33: 21810–21823, 2020.

Kumar, A., Fu, J., Soh, M., Tucker, G., and Levine, S. Stabilizing off-policy q-learning via bootstrapping error reduction. *Advances in Neural Information Processing Systems*, 32, 2019.

Kumar, A., Zhou, A., Tucker, G., and Levine, S. Conservative q-learning for offline reinforcement learning. *Advances in Neural Information Processing Systems*, 33: 1179–1191, 2020.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Levine, S., Kumar, A., Tucker, G., and Fu, J. Offline reinforcement learning: Tutorial, review, and perspectives on open problems. *arXiv preprint arXiv:2005.01643*, 2020.

Li, J., Vuong, Q., Liu, S., Liu, M., Ciosek, K., Christensen, H., and Su, H. Multi-task batch reinforcement learning with metric learning. *Advances in Neural Information Processing Systems*, 33:6197–6210, 2020a.

Li, L., Yang, R., and Luo, D. Focal: Efficient fully-offline meta-reinforcement learning via distance metric learning and behavior regularization. *arXiv preprint arXiv:2010.01112*, 2020b.

Liu, Y., Swaminathan, A., Agarwal, A., and Brunskill, E. Provably good batch reinforcement learning without great exploration. *arXiv preprint arXiv:2007.08202*, 2020.

Lu, C., Ball, P. J., Parker-Holder, J., Osborne, M. A., and Roberts, S. J. Revisiting design choices in offline model-based reinforcement learning. *arXiv preprint arXiv:2110.04135*, 2021.

Mendonca, R., Geng, X., Finn, C., and Levine, S. Meta-reinforcement learning robust to distributional shift via model identification and experience relabeling. *arXiv preprint arXiv:2006.07178*, 2020.

Mitchell, E., Rafailov, R., Peng, X. B., Levine, S., and Finn, C. Offline meta-reinforcement learning with advantage weighting. In *International Conference on Machine Learning*, pp. 7780–7791. PMLR, 2021.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529, 2015.

Nair, A., Gupta, A., Dalal, M., and Levine, S. Awac: Accelerating online reinforcement learning with offline datasets. *arXiv preprint arXiv:2006.09359*, 2020.

Pathak, D., Agrawal, P., Efros, A. A., and Darrell, T. Curiosity-driven exploration by self-supervised prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pp. 16–17, 2017.

Peng, X. B., Kumar, A., Zhang, G., and Levine, S. Advantage-weighted regression: Simple and scalable off-policy reinforcement learning. *arXiv preprint arXiv:1910.00177*, 2019.

Pong, V. H., Nair, A. V., Smith, L. M., Huang, C., and Levine, S. Offline meta-reinforcement learning with online self-supervision. In *International Conference on Machine Learning*, pp. 17811–17829. PMLR, 2022.

Rafailov, R., Yu, T., Rajeswaran, A., and Finn, C. Offline reinforcement learning from images with latent space models. In *Learning for Dynamics and Control*, pp. 1154–1168. PMLR, 2021.

Rakelly, K., Zhou, A., Finn, C., Levine, S., and Quillen, D. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In *International Conference on Machine Learning*, pp. 5331–5340, 2019.

Ren, T., Li, J., Dai, B., Du, S. S., and Sanghavi, S. Nearly horizon-free offline reinforcement learning. *Advances in neural information processing systems*, 34, 2021.

Ren, Z., Liu, A., Liang, Y., Peng, J., and Ma, J. Efficient meta reinforcement learning for preference-based fast adaptation. *arXiv preprint arXiv:2211.10861*, 2022.

Shi, L., Li, G., Wei, Y., Chen, Y., and Chi, Y. Pessimistic q-learning for offline reinforcement learning: Towards optimal sample complexity. *arXiv preprint arXiv:2202.13890*, 2022.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. Mastering the game of go without human knowledge. *nature*, 550(7676):354–359, 2017.

Smallwood, R. D. and Sondik, E. J. The optimal control of partially observable markov processes over a finite horizon. *Operations research*, 21(5):1071–1088, 1973.

Strens, M. A bayesian framework for reinforcement learning. In *ICML*, volume 2000, pp. 943–950, 2000.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Szepesvári, C. Lecture 17: Batch rl: Introduction discussion. In *CMPUT 653: Theoretical Foundations of Reinforcement Learning, University of Alberta*. 2022. OpenCourseLecture.

Wang, J. X., Kurth-Nelson, Z., Tirumala, D., Soyer, H., Leibo, J. Z., Munos, R., Blundell, C., Kumaran, D., and Botvinick, M. Learning to reinforcement learn. *arXiv preprint arXiv:1611.05763*, 2016.

Wu, Y., Tucker, G., and Nachum, O. Behavior regularized offline reinforcement learning. *arXiv preprint arXiv:1911.11361*, 2019.

Yin, M. and Wang, Y.-X. Towards instance-optimal offline reinforcement learning with pessimism. *Advances in neural information processing systems*, 34, 2021.

Yin, M., Bai, Y., and Wang, Y.-X. Near-optimal provable uniform convergence in offline policy evaluation for reinforcement learning. *arXiv preprint arXiv:2007.03760*, 2020.

Yin, M., Bai, Y., and Wang, Y.-X. Near-optimal offline reinforcement learning via double variance reduction. *Advances in neural information processing systems*, 34, 2021.

Yu, F., Xian, W., Chen, Y., Liu, F., Liao, M., Madhavan, V., and Darrell, T. Bdd100k: A diverse driving video database with scalable annotation tooling. *arXiv preprint arXiv:1805.04687*, 2(5):6, 2018.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., and Levine, S. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning*, pp. 1094–1100. PMLR, 2020a.

Yu, T., Quillen, D., He, Z., Julian, R., Hausman, K., Finn, C., and Levine, S. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on robot learning*, pp. 1094–1100. PMLR, 2020b.

Yu, T., Thomas, G., Yu, L., Ermon, S., Zou, J. Y., Levine, S., Finn, C., and Ma, T. Mopo: Model-based offline policy optimization. *Advances in Neural Information Processing Systems*, 33:14129–14142, 2020c.

Yuan, H. and Lu, Z. Robust task representations for offline meta-reinforcement learning via contrastive learning. In *International Conference on Machine Learning*, pp. 25747–25759. PMLR, 2022.

Zhang, J., Wang, J., Hu, H., Chen, T., Chen, Y., Fan, C., and Zhang, C. Metacure: Meta reinforcement learning with empowerment-driven exploration. In *International Conference on Machine Learning*, pp. 12600–12610. PMLR, 2021.

Zhou, W., Bajracharya, S., and Held, D. Plas: Latent action space for offline reinforcement learning. *arXiv preprint arXiv:2011.07213*, 2020.

Zintgraf, L., Shiarlis, K., Igl, M., Schulze, S., Gal, Y., Hofmann, K., and Whiteson, S. Varibad: A very good method for bayes-adaptive deep rl via meta-learning. In *International Conference on Learning Representations*, 2019.

A. Theory

Our theory is the first to formalize the offline meta-RL with online adaptation using task-dependent behavior policies. We adopt the perspective of Bayesian RL to formalize task distribution, i.e., Bayes-Adaptive MDP (BAMDP) (Zintgraf et al., 2019), which is a popular theoretical framework for meta-RL. In this paper, we incorporate offline datasets with task-dependent behavior policies into BAMDPs and present a unique challenge: reward-transition distributional shift, which differs from state-action distributional shift in SMAC and single-task offline RL (Levine et al., 2020). The consistency between offline and online policy evaluation is a very important criterion to measure the efficiency of algorithms in offline RL (Levine et al., 2020). We find that filtering out out-of-distribution episodes in online adaptation can ensure the consistency of offline and online policy evaluation. Moreover, some insights are general for meta-RL. For example, Lemma 10 shows that, for a meta-testing task drawn from *arbitrary* task distribution, the distance from the closest meta-training task will asymptotically approach zero with high probability, as the number of sampled meta-training tasks grows.

A.1. Background

Throughout this paper, for a given non-negative integer $N \in \mathbb{Z}_+$, we use $[N]$ to denote the set $\{0, 1, \ldots, N-1\}$. For any object that is a function of/distribution over $\mathcal{S}$, $\mathcal{S} \times \mathcal{A}$, $\mathcal{S} \times \mathcal{A} \times \mathcal{S}$, or $\mathcal{S} \times \mathcal{A} \times \mathcal{R}$, we will treat it as a vector whenever convenient.

A.1.1. Finite-Horizon Single-Task RL

In single-task RL, an agent interacts with a Markov Decision Process (MDP) to maximize its cumulative reward (Sutton & Barto, 2018). A finite-horizon MDP is defined as a tuple $M = (\mathcal{S}, \mathcal{A}, \mathcal{R}, H, P, R)$ (Zintgraf et al., 2019; Du et al., 2019), where $\mathcal{S}$ is the state space, $\mathcal{A}$ is the action space, $\mathcal{R}$ is the reward space, $H \in \mathbb{Z}_+$ is the planning horizon, $P : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{S})$ is the transition function which takes a state-action pair and returns a distribution over states, and $R : \mathcal{S} \times \mathcal{A} \to \Delta(\mathcal{R})$ is the reward distribution. In particular, we consider finite state, action, and reward spaces in the theoretical analysis, i.e., $|\mathcal{S}| < \infty, |\mathcal{A}| < \infty, |\mathcal{R}| < \infty$. Without loss of generality, we assume a fixed initial state s_0[2]. A policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$ prescribes a distribution over actions for each state. The policy π induces a (random) H-horizon trajectory $\tau_H^\pi = (s_0, a_0, r_0, s_1, a_1, \ldots, s_{H-1}, a_{H-1}, r_{H-1})$, where $a_0 \sim \pi(s_0), r_0 \sim R(s_0, a_0), s_1 \sim P(s_0, a_0), a_1 \sim \pi(s_1)$, etc. To streamline our analysis, for each $h \in [H]$, we use $\mathcal{S}_h \subseteq \mathcal{S}$ to denote the set of states at h-th timestep, and we assume $\mathcal{S}_h$ do not intersect with each other. To simplify notation, we assume the transition from any state in S_{H-1} and any action to the initial state s_0, i.e., $\forall s \in S_{H-1}, a \in \mathcal{A}$, we have $P(s_0|s, a) = 1$[3]. We also assume $r_t \in [0, 1], \forall t \in [H]$ almost surely. Denote the probability of τ_H:

$$p(\tau_H^\pi) = \left(\prod_{t \in [H]} \pi(a_t|s_t) \cdot R(r_t|s_t, a_t) \right) \prod_{t \in [H-1]} P(s_{t+1}|s_t, a_t). \tag{6}$$

For any policy π, we define a value function $V_\pi : \mathcal{S} \to \mathbb{R}$ as: $\forall h \in [H], \forall s \in \mathcal{S}_h$,

$$\begin{aligned} V_\pi(s) &= \mathbb{E}_{s_h = s, a_t \sim \pi(\cdot|s_t), r_t \sim R(\cdot|s_t, a_t), s_{t+1} \sim P(\cdot|s_t, a_t)} \left[\sum_{t=h}^{H-1} r_t \right] \\ &= \begin{cases} \sum_{a \in \mathcal{A}} \pi(a|s) \cdot \mathbb{E}_{r \sim R(\cdot|s,a)}[r], & \text{if } h = H-1, \\ \sum_{a \in \mathcal{A}} \pi(a|s) \left(\mathbb{E}_{r \sim R(\cdot|s,a)}[r] + \sum_{s' \in \mathcal{S}_{h+1}} P(s'|s, a) V_\pi(s') \right), & \text{otherwise,} \end{cases} \end{aligned} \tag{7}$$

[2]Some papers assume the initial state is sampled from a distribution P_1. Note this is equivalent to assuming a fixed initial state s_0, by setting $P(s_0, a) = P_1$ for all $a \in \mathcal{A}$ and now our state s_1 is equivalent to the initial state in their assumption.

[3]The transition from the state in S_{H-1} does not affect learning in the finite-horizon MDP M.

and a visitation distribution of π is defined by $\rho_\pi(\cdot) : \Delta(\mathcal{S})$ which is $\forall h \in [H], \forall s \in \mathcal{S}_h$,

$$\rho_\pi(s) = \begin{cases} \frac{1}{H}, & \text{if } h = 0 \text{ and } s = s_0, \\ \sum_{\tilde{s} \in \mathcal{S}_{h-1}, \tilde{a} \in \mathcal{A}} \rho_\pi(\tilde{s}) \cdot \pi(\tilde{a}|\tilde{s}) \cdot P(s|\tilde{s}, \tilde{a}), & \text{if } h > 0, \\ 0, & \text{otherwise,} \end{cases} \tag{8}$$

and $\forall s \in \mathcal{S}, a \in \mathcal{A}, r \in \mathcal{R}$,

$$\rho_\pi(s, a) = \rho_\pi(s) \cdot \pi(a|s) \quad \text{and} \quad \rho_\pi(s, a, r) = \rho_\pi(s) \cdot \pi(a|s) \cdot R(r|s, a). \tag{9}$$

The expected total reward induced by policy π, i.e., the policy evaluation of π, is defined by

$$\mathcal{J}_M(\pi) = V_\pi(s_0) = H \sum_{s \in \mathcal{S}, a \in \mathcal{A}} \rho_\pi(s, a) \cdot \mathbb{E}_{r \sim R(\cdot|s,a)}[r]. \tag{10}$$

The goal of RL is to find a policy π that maximizes its expected return $\mathcal{J}(\pi)$.

A.1.2. Offline Finite-Horizon Single-Task RL

We consider the offline finite-horizon single-task RL setting, that is, a learner only has access to a dataset $\mathcal{D}$ consisting of K trajectories $\left\{\left(s_t^k, a_t^k, r_t^k\right)\right\}_{t \in [H]}^{k \in [K]}$ (i.e., $|\mathcal{D}| = KH$ tuples) and is not allowed to interact with the environment for additional online explorations. The data can be collected through multi-source logging policies and denote the unknown behavior policy μ. Similar with related work (Ren et al., 2021; Yin et al., 2020; Yin & Wang, 2021; Yin et al., 2021; Shi et al., 2022), we assume that $\mathcal{D}$ is collected through interacting K i.i.d. episodes using policy μ in M. Define the reward and transition distribution of data collection with μ in M by $\mathbb{P}_M$ (Jin et al., 2021), i.e., $\forall t \in [H]$ in each episode,

$$\mathbb{P}_M(r_t, s_{t+1} | s_t, a_t) = R^M(r_t | s_t, a_t) \cdot P^M(s_{t+1} | s_t, a_t), \tag{11}$$

where the action a_t is drawn from a behavior policy μ. Denote a dataset collected following the i.i.d. data collecting process, i.e., $\mathcal{D} \sim (\mathbb{P}_M, \mu)$ is an i.i.d. dataset. Note that the offline dataset $\mathcal{D}$ can be narrowly collected by some behavior policy μ and a large amount of state-action pairs are not contained in $\mathcal{D}$. These unseen state-action pairs will be erroneously estimated to have unrealistic values, called a phenomenon *extrapolation error* (Fujimoto et al., 2019). To overcome extrapolation error in policy learning of finite MDPs, Fujimoto et al. (Fujimoto et al., 2019) introduces batch-constrained RL, which restricts the action space in order to force policy selection of an agent with respect to a subset of the given data. Thus, define a batch-constrained policy set is

$$\Pi^{\mathcal{D}} = \{\pi | \pi(a|s) = 0 \text{ whenever } (s, a) \notin \mathcal{D}\}, \tag{12}$$

where denoting $(s, a) \in \mathcal{D}$ if there exists a trajectory containing (s, a) in the dataset $\mathcal{D}$, and similarly for $s \in \mathcal{D}$, $(s, a, r) \in \mathcal{D}$, or $(s, a, r, s') \in \mathcal{D}$. The batch-constrained policy set $\Pi^{\mathcal{D}}$ consists of the policies that for any state s observed in the dataset $\mathcal{D}$, the agent will not select an action outside of the dataset. Thus, for any batch-constrained policy $\pi \in \Pi^{\mathcal{D}}$, define the approximate value function $V_\pi^{\mathcal{D}} : \mathcal{S} \to \mathbb{R}$ estimated from $\mathcal{D}$ (Fujimoto et al., 2019; Liu et al., 2020) as: $\forall h \in [H], \forall s \in \mathcal{S}_h$,

$$V_\pi^{\mathcal{D}}(s) = \mathbb{E}_{s_h = s, a_t \sim \pi(\cdot|s_t), (s_t, a_t, r_t, s_{t+1}) \sim \mathcal{D}} \left[\sum_{t=h}^{H-1} r_t\right] \tag{13}$$

$$= \begin{cases} \sum_{a \in \mathcal{A}} \pi(a|s) \mathbb{E}_{(s,a,r) \in \mathcal{D}}[r], & \text{if } h = H - 1, \\ \sum_{a \in \mathcal{A}} \pi(a|s) \mathbb{E}_{(s,a,r,s') \in \mathcal{D}}\left[r + V_\pi^{\mathcal{D}}(s')\right], & \text{otherwise,} \end{cases} \tag{14}$$

which is called Approximate Dynamic Programming (ADP) (Bertsekas & Tsitsiklis, 1995) and such methods take sampling data as input and approximate the value-function (Liu et al., 2020; Chen & Jiang, 2019). In addition, define the approximate policy evaluation of π estimated from $\mathcal{D}$ as

$$\mathcal{J}_{\mathcal{D}}(\pi) = V_\pi^{\mathcal{D}}(s_0). \tag{15}$$

The offline RL literature (Fujimoto et al., 2019; Liu et al., 2020; Chen & Jiang, 2019; Kumar et al., 2019; 2020) aims to utilize approximate expected total reward $\mathcal{J}_{\mathcal{D}}(\pi)$ with various conservatism regularizations (i.e., policy constraints, policy penalty, uncertainty penalty, etc.) (Levine et al., 2020) to find a good policy within a batch-constrained policy set $\Pi^{\mathcal{D}}$.

Similar to offline finite-horizon single-task RL theory (Ren et al., 2021; Yin et al., 2020; Yin & Wang, 2021; Yin et al., 2021; Shi et al., 2022), define

$$d_{\mu}^{M} = \min\left\{\rho_{\mu}(s,a)\,|\rho_{\mu}(s,a) > 0, \forall s \in \mathcal{S}, a \in \mathcal{A}\right\}, \tag{16}$$

which is the minimal visitation state-action distribution induced by the behavior policy μ in M and is an intrinsic quantity required by theoretical offline learning (Yin et al., 2020). Note that, different from recent offline episodic RL theory (Ren et al., 2021; Yin et al., 2020; Yin & Wang, 2021; Yin et al., 2021; Shi et al., 2022), we do not assume any weak or uniform coverage assumption in the dataset because we focus on the policy evaluation of all batch-constrained policies in $\Pi^{\mathcal{D}}$ rather than the optimal policy in the MDP M.

A.1.3. Standard meta-RL

The goal of meta-RL (Finn et al., 2017; Rakelly et al., 2019) is to train a meta-policy that can quickly adapt to new tasks using N adaptation episodes. The standard meta-RL setting deals with a distribution $p(\kappa)$ over MDPs, in which each task κ_i sampled from $p(\kappa)$ presents a finite-horizon MDP (Zintgraf et al., 2019; Du et al., 2019). κ_i is defined by a tuple $(\mathcal{S}, \mathcal{A}, \mathcal{R}, H, P^{\kappa_i}, R^{\kappa_i})$, including state space $\mathcal{S}$, action space $\mathcal{A}$, reward space $\mathcal{R}$, planning horizon H, transition function $P^{\kappa_i}(s'|s,a)$, and reward function $R^{\kappa_i}(r|s,a)$. Denote $\mathcal{K}$ is the space of task κ_i. In this paper, we assume dynamics function P and reward function R may vary across tasks and share a common structure. The meta-RL algorithms repeatedly sample batches of tasks to train a meta-policy. In the meta-testing, agents aim to rapidly adapt a good policy for new tasks drawn from $p(\kappa)$.

POMDPs. We can formalize the meta-RL with few-shot adaptation as a specific finite-horizon Partially Observable Markov Decision Process (POMDP), which is defined by a tuple $\widehat{M} = \left(\widehat{\mathcal{S}}, \mathcal{A}, \mathcal{R}, \Omega, \widehat{H}, \widehat{P}, \widehat{P}_0, O, \widehat{R}\right)$, where $\widehat{\mathcal{S}} = \mathcal{S} \times \mathcal{K}$ is the state space, $\mathcal{A}$ and $\mathcal{R}$ are the same action and reward spaces as the finite-horizon MDP M defined in Appendix A.1.1, respectively, $\Omega = \mathcal{S}$ is the observation space, $\widehat{H} = N \times H$ is the planning horizon which represents N adaptation episodes for a single meta-RL MDP κ_i, as discussed in Zintgraf et al. (Zintgraf et al., 2019), $\widehat{P} : \widehat{\mathcal{S}} \times \mathcal{A} \to \Delta\left(\widehat{\mathcal{S}}\right)$ is the transition function: $\forall \hat{s}, \hat{s}' \in \widehat{\mathcal{S}}, a \in \mathcal{A}$, where denoting $\hat{s} = (s, \kappa_i)$ and $\hat{s}' = (s', \kappa_j)$,

$$\widehat{P}\left(\hat{s}'|\hat{s}, a\right) = \begin{cases} P^{\kappa_i}(s'|s,a), & \text{if } \kappa_i = \kappa_j, \\ 0, & \text{otherwise}, \end{cases} \tag{17}$$

$\widehat{P}_0 : \Delta\left(\widehat{\mathcal{S}}\right)$ is the initial state distribution: $\forall \hat{s} = (s, \kappa_i) \in \widehat{\mathcal{S}}$,

$$\widehat{P}_0\left(\hat{s}\right) = \begin{cases} p(\kappa_i), & \text{if } s = s_0, \\ 0, & \text{otherwise}, \end{cases} \tag{18}$$

$O : \widehat{\mathcal{S}} \to \Delta\left(\Omega\right)$ is the observation probability distribution conditioned on a state: $\forall \hat{s} = (s, \kappa_i) \in \widehat{\mathcal{S}}, o \in \Omega$,

$$O\left(o|\hat{s}\right) = \begin{cases} 1, & \text{if } o = s, \\ 0, & \text{otherwise}, \end{cases} \tag{19}$$

and $\widehat{R} : \widehat{\mathcal{S}} \times \mathcal{A} \to \Delta\left(\mathcal{R}\right)$ is the reward distribution: $\forall \hat{s} = (s, \kappa_i) \in \widehat{\mathcal{S}}, a \in \mathcal{A}, r \in \mathcal{R}$,

$$\widehat{R}\left(r|\hat{s}, a\right) = R^{\kappa_i}(r|s,a). \tag{20}$$

Denote context $c_t = (a_t, r_t, s_{t+1})$ as an experience collected at timestep t, and $c_{:t} = \langle s_0, c_0, \ldots, c_{t-1}\rangle$[4]$\in \mathcal{C}_t \equiv \Omega \times (\mathcal{A} \times \mathcal{R} \times \Omega)^t$ indicates all experiences collected during t timesteps. Note that t may be larger than H, and when it is the

[4]For clarity, we denote $c_{:0}^{\kappa_i} = s_0$.

case, $c_{:t}$ represents experiences collected across episodes in the single meta-RL MDP κ_i. Denote the entire context space $\mathcal{C} = \bigcup_{t=0}^{\widehat{H}-1} \mathcal{C}_t$ and a meta-policy $\hat{\pi} : \mathcal{C} \to \Delta(\mathcal{A})$ (Wang et al., 2016; Duan et al., 2016) prescribes a distribution over actions for each context. The goal of meta-RL is to find a meta-policy on history contexts $\hat{\pi}$ that maximizes the expected return within N adaptation episodes:

$$\mathcal{J}_{\widehat{M}}(\hat{\pi}) = \mathbb{E}_{\hat{s}_0 \sim P_0, o_t \sim O(\cdot|s_t), a_t \sim \hat{\pi}(\cdot|c_{:t}), r_t \sim \widehat{R}(\cdot|s_t,a_t), \hat{s}_{t+1} \sim \widehat{P}(\cdot|\hat{s}_t,a_t)} \left[\sum_{t=0}^{\widehat{H}-1} r_t \right] \tag{21}$$

$$= \mathbb{E}_{\kappa_i \sim p(\kappa)} \left[\sum_{j=0}^{N-1} \mathbb{E}_{a_t \sim \hat{\pi}\left(\cdot|c_{:(jH+t)}\right), r_t \sim R^{\kappa_i}(\cdot|s_t,a_t), s_{t+1} \sim P^{\kappa_i}(\cdot|s_t,a_t)} \left[\sum_{t=0}^{H-1} r_t \right] \right]. \tag{22}$$

BAMDPs. A Markovian belief state allows a POMDP to be formulated as a Markov decision process where every belief is a state (Cassandra et al., 1994). We can transform the finite-horizon POMDP $\widehat{M}$ to a finite-horizon belief MDP, which is called Bayes-Adaptive MDP (BAMDP) in the literature (Zintgraf et al., 2019; Ghavamzadeh et al., 2015; Dorfman et al., 2021) and is defined by a tuple $M^+ = \left(\mathcal{S}^+, \mathcal{A}, \mathcal{R}, H^+, P^+, P_0^+, R^+\right)$, $\mathcal{S}^+ = \mathcal{S} \times \mathcal{B}$ is the hyper-state space, where $\mathcal{B} = \{p(\kappa|c) \,|c \in \mathcal{C}\}$ is the set of task beliefs over the meta-RL MDPs, the prior

$$b_0^\kappa = p\left(\kappa|c_{:0}\right) = p(\kappa) \tag{23}$$

is the meta-RL MDP distribution, and $\forall t \in \left[\widehat{H}-1\right]$, $\forall c_{:(t+1)} \in \mathcal{C}$, denoting $b_t^\kappa = p\left(\kappa|c_{:t}\right)$ and

$$b_{t+1}^\kappa = p\left(\kappa|c_{:(t+1)}\right) = p\left(p\left(\kappa|c_{:t}\right)|c_{:(t+1)}\right) = p\left(p\left(\kappa|c_{:t}\right)|s_t, c_t\right) = p\left(b_t^\kappa|s_t, c_t\right) \tag{24}$$

$$\propto p\left(b_t^\kappa, c_t|s_t\right) = p\left(c_t|s_t, b_t^\kappa\right) p\left(b_t^\kappa|s_t\right) = p\left(c_t|s_t, b_t^\kappa\right) b_t^\kappa \tag{25}$$

$$= \mathbb{E}_{\kappa_i \sim b_t^\kappa}\left[R^{\kappa_i}(r_t|s_t,a_t) \cdot P^{\kappa_i}(s_{t+1}|s_t,a_t)\right] \cdot b_t^\kappa \tag{26}$$

is the posterior over the MDPs given the context $c_{:(t+1)}$, $\mathcal{A}, \mathcal{R}$ are the same action space and reward space as the finite-horizon POMDP $\widehat{M}$, respectively, $H^+ = N \times H$ is the planning horizon across adaptation episodes, $P^+ : \mathcal{S}^+ \times \mathcal{A} \times \mathcal{R} \to \Delta(\mathcal{S}^+)$ is the transition function: $\forall s_t^+, s_{t+1}^+ \in \mathcal{S}^+, a_t \in \mathcal{A}, r_t \in \mathcal{R}$, where denoting $s_t^+ = (s_t, b_t^\kappa)$ and $s_{t+1}^+ = \left(s_{t+1}, \tilde{b}_{t+1}^\kappa\right)$,

$$P^+\left(s_{t+1}^+ \middle| s_t^+, a_t, r_t\right) = P^+\left(s_{t+1}, \tilde{b}_{t+1}^\kappa \,|s_t, b_t^\kappa, a_t, r_t\right) \tag{27}$$

$$= P^+\left(s_{t+1}\,|s_t, b_t^\kappa, a_t\right) P^+\left(\tilde{b}_{t+1}^\kappa\,|s_t, b_t^\kappa, c_t\right) \tag{28}$$

$$= \mathbb{E}_{\kappa_i \sim b_t^\kappa}\left[P^{\kappa_i}(s_{t+1}|s_t,a_t)\right] \cdot \mathbb{1}\left[\tilde{b}_{t+1}^\kappa = p(b_t^\kappa|s_t,c_t)\right], \tag{29}$$

$P_0^+ : \Delta(\mathcal{S}^+)$ is the initial hyper-state distribution, i.e., a deterministic initial hyper-state is

$$s_0^+ = (s_0, b_0^\kappa) = (s_0, p(\kappa)) \in \mathcal{S}^+, \tag{30}$$

and $R^+ : \mathcal{S}^+ \times \mathcal{A} \to \Delta(\mathcal{R})$ is the reward distribution: $\forall s^+ = (s, b^\kappa) \in \mathcal{S}^+, a \in \mathcal{A}, r \in \mathcal{R}$,

$$R^+\left(r|s^+, a\right) = R^+\left(r|s, b^\kappa, a\right) = \mathbb{E}_{\kappa_i \sim b^\kappa}\left[R^{\kappa_i}(r|s,a)\right]. \tag{31}$$

In a BAMDP, the belief is over the transition and reward functions, which are constant for a given task. A meta-policy on BAMDP $\pi^+ : \mathcal{S}^+ \to \Delta(\mathcal{A})$ prescribes a distribution over actions for each hyper-state. The agent's objective is now to find a meta-policy on hyper-states π^+ that maximizes the expected return in the BAMDP,

$$\mathcal{J}_{M^+}\left(\pi^+\right) = \mathbb{E}_{a_t \sim \pi^+\left(\cdot|s_t^+\right), r_t \sim R^+\left(\cdot|s_t^+, a_t\right), s_{t+1}^+ \sim P^+\left(\cdot|s_t^+, a_t\right)} \left[\sum_{t=0}^{\widehat{H}-1} r_t \right] \tag{32}$$

$$= \mathbb{E}_{\kappa_i \sim p(\kappa)} \left[\sum_{j=0}^{N-1} \mathbb{E}_{a_t \sim \pi^+\left(\cdot|s_{jH+t}^+\right), r_t \sim R^{\kappa_i}(\cdot|s_t,a_t), s_{t+1} \sim P^{\kappa_i}(\cdot|s_t,a_t)} \left[\sum_{t=0}^{H-1} r_t \right] \right]. \tag{33}$$

For any meta-policy on hyper-states π^+, denote the corresponding meta-policy on history contexts $\hat{f}_{\pi^+} : \mathcal{C} \to \Delta(\mathcal{A})$, i.e., $\forall t \in \left[\widehat{H} - 1\right], \forall c_{:t} \in \mathcal{C}_t$, s.t., $\hat{f}_{\pi^+}(\cdot|c_{:t}) = \pi^+\left(\cdot|s_t^+\right)$, where $s_t^+ = (s_t, b_t^\kappa) = (s_t, p(\kappa|c_{:t}))$, and we have

$$\mathcal{J}_{\widehat{M}}\left(\hat{f}_{\pi^+}\right) = \mathbb{E}_{\kappa_i \sim p(\kappa)}\left[\sum_{j=0}^{N-1} \mathbb{E}_{a_t \sim \hat{f}_{\pi^+}\left(\cdot\left|c_{:(jH+t)}\right.\right), r_t \sim R^{\kappa_i}(\cdot|s_t,a_t), s_{t+1} \sim P^{\kappa_i}(\cdot|s_t,a_t)}\left[\sum_{t=0}^{H-1} r_t\right]\right] \tag{34}$$

$$= \mathcal{J}_{M^+}\left(\pi^+\right). \tag{35}$$

The belief MDP is such that an optimal policy for it, coupled with the correct state estimator, will give rise to optimal behavior for the original POMDP (Astrom, 1965; Smallwood & Sondik, 1973; Kaelbling et al., 1998), which indicates that

$$\mathcal{J}_{M^+}\left(\pi^{+,*}\right) = \mathcal{J}_{\widehat{M}}\left(\hat{f}_{\pi^{+,*}}\right) = \mathcal{J}_{\widehat{M}}\left(\hat{\pi}^*\right), \tag{36}$$

where $\pi^{+,*}$ and $\hat{\pi}^*$ are the optimal policies for BAMDP M^+ and POMDP $\widehat{M}$, respectively. Thus, the agent can find a policy π^+ to maximize the expected return in the BAMDP M^+ to address the POMDP $\widehat{M}$ by the transformed policy $\hat{f}_{\pi^+}$.

A.1.4. Offline meta-RL

In the offline meta-RL setting, a meta-learner only has access to an offline multi-task dataset $\mathcal{D}^+$ and is not allowed to interact with the environment during meta-training (Li et al., 2020b). Recent offline meta-RL methods (Dorfman et al., 2021) always utilize task-dependent behavior policies $p(\mu|\kappa)$, which represents the random variable of the behavior policy $\mu(a|s)$ conditioned on the random variable of the task κ. For brevity, we overload $[\mu] = p(\mu|\kappa)$. Similar to related work on offline RL (Shi et al., 2022), we assume that $\mathcal{D}^+$ is collected through interacting multiple i.i.d. trajectories using task-dependent policies $[\mu]$ in M^+. Define the reward and transition distribution of the task-dependent data collection by $\mathbb{P}_{M^+,[\mu]}$ (Jin et al., 2021), i.e., for each step t in a trajectory,

$$\mathbb{P}_{M^+,[\mu]}\left(r_t, s_{t+1}\left|s_t^+, a_t\right.\right) \propto \mathbb{E}_{\kappa_i \sim p(\kappa), \mu_i \sim p(\mu|\kappa_i)}\left[\mathbb{P}_{\kappa_i}\left(r_t, s_{t+1}\left|s_t, a_t\right.\right) \cdot p_{M^+}\left(s_t^+\left|\kappa_i, \mu_i\right.\right)\right], \tag{37}$$

where $\mathbb{P}_{\kappa_i}$ is the reward and transition distribution of κ_i defined in Eq. (11), and $p_{M^+}\left(s_t^+\left|\kappa_i, \mu_i\right.\right)$ denotes the probability of s_t^+ when executing μ_i in a task κ_i, i.e.,

$$p_{M^+}\left(s_t^+\left|\kappa_i, \mu_i\right.\right) = \sum_{c_{:t} \in \mathcal{C}_t} p_{\kappa_i}^{\mu_i}(c_{:t}) \cdot \mathbb{1}\left[b_t^\kappa = p(\kappa|c_{:t})\right], \tag{38}$$

where the state in $c_{:t}$ is s_t and $p_{\kappa_i}^{\mu_i}(c_{:t})$ is defined in Eq. (6). Similar to offline single-task RL (see Appendix A.1.2), offline dataset $\mathcal{D}^+$ can be narrow and a large amount of state-action pairs are not contained. These unseen state-action pairs will be erroneously estimated to have unrealistic values, called a phenomenon *extrapolation error* (Fujimoto et al., 2019). To overcome extrapolation error in offline RL, related works (Fujimoto et al., 2019) introduce batch-constrained RL, which restricts the action space in order to force policy selection of an agent with respect to a given dataset. Define a policy π^+ to be batch-constrained by $\mathcal{D}^+$ if $\pi^+\left(a\left|s^+\right.\right) = 0$ whenever a tuple (s^+, a) is not contained in $\mathcal{D}^+$. Offline RL (Liu et al., 2020; Chen & Jiang, 2019) approximates policy evaluation for a batch-constrained policy π^+ by sampling from an offline dataset $\mathcal{D}^+$, which is denoted by $\mathcal{J}_{\mathcal{D}^+}\left(\pi^+\right)$ and called *Approximate Dynamic Programming* (ADP; Bertsekas & Tsitsiklis, 1995). During meta-testing, RL agents perform online adaptation using a meta-policy π^+ in new tasks drawn from meta-RL task distribution. The reward and transition distribution of data collection with π^+ in M^+ during adaptation is defined by

$$\mathbb{P}_{M^+,\pi^+}\left(r_t, s_{t+1}\left|s_t^+, a_t\right.\right) = R^+\left(r_t\left|s_t^+, a_t\right.\right) \cdot P^+\left(s_{t+1}\left|s_t^+, a_t\right.\right), \tag{39}$$

where $P^+\left(s_{t+1}\left|s_t^+, a_t\right.\right)$ is the marginal transition functions of M^+, i.e.,

$$P^+\left(s_{t+1}\left|s_t^+, a_t\right.\right) = \mathbb{E}_{\kappa_i \sim b_t^\kappa}\left[P^{\kappa_i}\left(s_{t+1}\left|s_t, a_t\right.\right)\right]. \tag{40}$$

A.2. Main Results in Section 3.1

Definition 1 (Transition-Reward Distribution Shift)**.** *In a BAMDP M^+, for each task-dependent behavior policy $[\mu]$ and batch-constrained meta-policy π^+, the transition-reward distribution shift is defined by that there exists a pair of $\left(s_t^+, a_t\right)$ with executing π^+ in M^+, s.t.,*

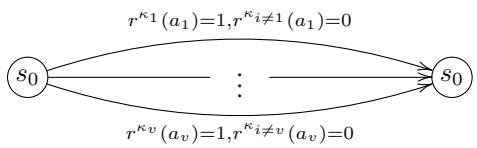
$$\mathbb{P}_{M^+,[\mu]}\left(r_t, s_{t+1} \left| s_t^+, a_t\right.\right) \neq \mathbb{P}_{M^+,\pi^+}\left(r_t, s_{t+1} \left| s_t^+, a_t\right.\right), \tag{1}$$

where $\mathbb{P}_{M^+,[\mu]}, \mathbb{P}_{M^+,\pi^+}$ are the reward and transition distribution of offline data collection by $[\mu]$ and online data collection by π^+, respectively, whose formal definition are deferred to Appendix A.1.4.

This definition utilizes the discrepancy between offline and online data collection to characterize the joint distribution gap of reward and transition. Note that in offline data collection $\mathbb{P}_{M^+,[\mu]}$, the behavior policies $p(\mu|\kappa)$ can vary based on task identification, whereas the online data collection $\mathbb{P}_{M^+,\pi^+}$ is the expected reward and transition distribution across the task distribution $p(\kappa)$. Formally, we use $p_{M^+}^{\pi^+}\left(s_t^+, a_t\right) > 0$ to present that $\left(s_t^+, a_t\right)$ can be reached by π^+ in M^+, where

$$p_{M^+}^{\pi^+}\left(s_t^+, a_t\right) = p_{M^+}^{\pi^+}(s_t^+) \cdot \pi^+\left(a_t \left| s_t^+\right.\right), \tag{41}$$

and $p_{M^+}^{\pi^+}(s_t^+)$ is defined in Eq. (6). The data distribution induced by π^+ and $[\mu]$ mismatches when the reward and transition distribution of π^+ and $[\mu]$ differs in a tuple $\left(s_t^+, a_t\right)$, in which the agent can reach this tuple by executing π^+ in M^+, i.e., $p_{M^+}^{\pi^+}\left(s_t^+, a_t\right) > 0$. Note that if π^+ can reach a tuple $\left(s_t^+, a_t\right)$, this tuple is guaranteed to be contained in the offline dataset, i.e., $p_{M^+}^{[\mu]}\left(s_t^+, a_t\right) > 0$, because a batch-constrained policy π^+ will not select an action outside of the dataset collected by $[\mu]$, as introduced in Section 2.2.



$$p(\kappa_i) = \tfrac{1}{v}, p(\mu_i|\kappa_i) = 1, \text{and } \mu_i(a_i|s_0) = 1$$

Figure 4. A concrete example, which has v meta-RL tasks, one state, v actions, v behavior policies, horizon $H = 1$ in a episode, and v adaptation episodes, where $v \geq 3$.

Theorem 1. *There exists a BAMDP M^+ with task-dependent behavior policies $[\mu]$ such that, for any batch-constrained meta-policy π^+, the transition-reward distribution shift between $\mathbb{P}_{M^+,[\mu]}$ and $\mathbb{P}_{M^+,\pi^+}$ occurs.*

Proof. To serve a concrete example, we construct an offline meta-RL setting shown in Figure 4. In this example, there are v meta-RL tasks $\mathcal{K} = \{\kappa_1, \ldots, \kappa_v\}$ and v behavior policies $\{\mu_1, \ldots, \mu_v\}$, where $v \geq 3$. Each task κ_i has one state $\mathcal{S} = \{s_0\}$, $2v$ actions $\mathcal{A} = \{a_1, \ldots, a_v\}$, and horizon in an episode $H = 1$. For each task κ_i, RL agents can receive reward 1 performing action a_i. During adaptation, the RL agent can interact with the environment within v episodes. The task distribution is uniform, the behavior policy of task κ_i is μ_i, and each behavior policy μ_i will perform a_i. When a batch-constrained meta-policy π^+ selects an action $\tilde{a}$ in the initial state s_0^+, we find that

$$\mathbb{P}_{M^+,[\mu]}\left(r = 1 \left| s_0^+, \tilde{a}\right.\right) = 1 \neq \mathbb{P}_{M^+,\pi^+}\left(r = 1 \left| s_0^+, \tilde{a}\right.\right) = \frac{1}{v}, \tag{42}$$

in which there is the probability of $\frac{1}{v}$ to sample a corresponding testing task, whose reward function of $\tilde{a}$ is 1, whereas the reward in the offline dataset collected by $[\mu]$ is all 1. □

A.3. Main Results in Section 3.2

A.3.1. Out-of-Distribution Analyses

Proposition 1. *There exists a BAMDP M^+ with task-dependent behavior policies such that, for any batch-constrained meta-policy π^+, (i) RL agents will visit out-of-distribution hyper-states and (ii) the gap between offline policy evaluation $\mathcal{J}_{\mathcal{D}^+}(\pi^+)$ and online policy evaluation $\mathcal{J}_{M^+}(\pi^+)$ is at least $\frac{H^+-1}{2}$.*

Proof. **Part (i)** In the example shown in Figure 4, an offline multi-task dataset D^+ is drawn from the task-dependent data collection $\mathbb{P}_{M^+,[\mu]}$. Since the reward of D^+ is all 1, the task beliefs in D^+ have two types: (i) all task possible s_0^+ and (ii) determining task i with receiving reward 1 in action a_i. For any batch-constrained meta-policy π^+ selecting an action $\tilde{a}_j$ on s_0^+ during meta-testing, there has probability $1-\frac{1}{v}$ to receive reward 0 and the task belief will become "excluding task j", which is not contained in D^+ with $v \geq 3$. For any $\delta \in (0,1]$, let $v > \frac{1}{\delta}$, with probability $1-\delta$, the agent will visit out-of-distribution hyper-states during adaptation.

Part (ii) In Figure 4, an offline dataset D^+ only contains reward 1, thus for each batch-constrained meta-policy π^+, the offline evaluation of π^+ in D^+ is $\mathcal{J}_{D^+}(\pi^+) = H^+ = vH$. The optimal meta-policy $\pi^{+,*}$ in this example is to enumerate $a_1, \ldots, a_v$ until the task identification is inferred from an action with a reward of 1. A meta-policy $\pi^{+,*}$ needs to explore in the testing environments and its online policy evaluation is

$$\mathcal{J}_{M^+}\left(\pi^{+,*}\right) = \sum_{k=0}^{N-1} \frac{N-k}{v-k} \prod_{j=0}^{k-1} \left(1 - \frac{1}{v-j}\right) \tag{43}$$

$$= \sum_{k=0}^{N-1} \prod_{j=0}^{k-1} \frac{v-j-1}{v-j} \tag{44}$$

$$= \sum_{k=0}^{N-1} \frac{v-k}{v} = N - \frac{N(N-1)}{2v} \tag{45}$$

$$= \frac{v+1}{2} = \frac{H^+ + 1}{2}, \tag{46}$$

where $N = v$ is the number of adaptation episodes. Thus, the gap of policy evaluation of π^+ between offline meta-training and online adaptation is

$$\left|\mathcal{J}_{M^+}\left(\pi^+\right) - \mathcal{J}_{D^+}\left(\pi^+\right)\right| \geq \mathcal{J}_{D^+}\left(\pi^+\right) - \mathcal{J}_{M^+}\left(\pi^{+,*}\right) = \frac{H^+ - 1}{2}. \tag{47}$$

□

Proposition 1 states that RL agents will go out of the distribution of the offline dataset $\mathcal{D}^+$ due to shifts in the reward and transition distribution. Thus, the offline policy evaluation of π^+ in meta-training cannot provide a reference for the online meta-testing.

A.3.2. In-Distribution Analyses

Definition 2 (Transformed BAMDPs)**.** *A transformed BAMDP is defined as a tuple $\overline{M}^+ = \left(\overline{\mathcal{S}}^+, \mathcal{A}, \mathcal{R}, H^+, \overline{P}^+, \overline{P}_0^+, \overline{R}^+\right)$, where $\overline{\mathcal{S}}^+ = \mathcal{S} \times \overline{\mathcal{B}}$ is the hyper-state space, $\overline{\mathcal{B}}$ is the space of overall beliefs over meta-RL MDPs with behavior policies, $\mathcal{A}, \mathcal{R}, H^+$ are the same action space, reward space, and planning horizon as the original BAMDP M^+, respectively, $\overline{P}_0^+$ is the initial hyper-state distribution presenting joint distribution of task and behavior policies $p(\kappa,\mu) = p(\kappa)p(\mu|\kappa)$, and $\overline{P}^+, \overline{R}^+$ are the transition and reward functions. The goal of meta-RL agents is to find a meta-policy $\bar{\pi}^+\left(a_t \left| \bar{s}_t^+\right.\right)$ that maximizes online policy evaluation $\mathcal{J}_{\overline{M}^+}(\bar{\pi}^+)$. Denote the reward and transition distribution of the task-dependent data collection in a transformed BAMDP $\overline{M}^+$ by $\mathbb{P}_{\overline{M}^+,[\mu]}$, as defined in Eq. (37). Denote the offline multi-task dataset collected by task-dependent data collection $\mathbb{P}_{\overline{M}^+,[\mu]}$ by $\overline{\mathcal{D}}^+$.*

More specifically, a finite-horizon transformed BAMDP is defined by a tuple $\overline{M}^+ = \left(\overline{\mathcal{S}}^+, \mathcal{A}, \mathcal{R}, H^+, \overline{P}^+, \overline{P}_0^+, \overline{R}^+\right)$, $\overline{\mathcal{S}}^+ = \mathcal{S} \times \overline{\mathcal{B}}$ is the hyper-state space, where $\overline{\mathcal{B}} = \{p(\kappa,\mu|c) \,|c \in \mathcal{C}\}$ is the space of beliefs over meta-RL MDPs with behavior policies, the prior

$$b_0^{\kappa,\mu} = p\left(\kappa, \mu \left| c_{:0}\right.\right) = p(\kappa,\mu) \tag{48}$$

is the distribution of meta-RL MDPs with behavior policies, and $\forall t \in \left[\widehat{H}-1\right]$, $\forall c_{:(t+1)} \in \mathcal{C}$, denoting $b_t^{\kappa,\mu} = p\left(\kappa,\mu|c_{:t}\right)$ and

$$b_{t+1}^{\kappa,\mu} = p\left(\kappa,\mu \left| c_{:(t+1)}\right.\right) = p\left(p\left(\kappa,\mu|c_{:t}\right)\left|c_{:(t+1)}\right.\right) = p\left(p\left(\kappa,\mu|c_{:t}\right)|s_t,c_t\right) = p\left(b_t^{\kappa,\mu}\right| s_t, c_t\right) \tag{49}$$

$$\propto p\left(b_t^{\kappa,\mu}, c_t \middle| s_t\right) = p\left(c_t \middle| s_t, b_t^{\kappa,\mu}\right) p\left(b_t^{\kappa,\mu} \middle| s_t\right) = p\left(c_t \middle| s_t, b_t^{\kappa,\mu}\right) b_t^{\kappa,\mu} \tag{50}$$

$$= \mathbb{E}_{(\kappa_i,\mu_i)\sim b_t^{\kappa,\mu}}\left[\mu_i\left(a_t|s_t\right) \cdot R^{\kappa_i}(r_t|s_t,a_t) \cdot P^{\kappa_i}(s_{t+1}|s_t,a_t)\right] \cdot b_t^{\kappa,\mu} \tag{51}$$

is the posterior over the meta-RL MDPs with behavior policies given the context $c_{:(t+1)}$, $\mathcal{A}$, $\mathcal{R}$ and $\widehat{H}$ are the same action space, reward space, and planning horizon as the finite-horizon BAMDP M^+, respectively, $\overline{P}^+ : \overline{\mathcal{S}}^+ \times \mathcal{A} \times \mathcal{R} \to \Delta\left(\overline{\mathcal{S}}^+\right)$ is the transition function: $\forall \bar{s}_t^+, \bar{s}_{t+1}^+ \in \overline{\mathcal{S}}^+, a_t \in \mathcal{A}, r_t \in \mathcal{R}$, where denoting $\bar{s}_t^+ = (s_t, b_t^{\kappa,\mu})$ and $\bar{s}_{t+1}^+ = \left(s_{t+1}, \tilde{b}_{t+1}^{\kappa,\mu}\right)$,

$$\overline{P}^+\left(\bar{s}_{t+1}^+ \middle| \bar{s}_t^+, a_t, r_t\right) = \overline{P}^+\left(s_{t+1}, \tilde{b}_{t+1}^{\kappa,\mu} \middle| s_t, b_t^{\kappa,\mu}, a_t, r_t\right) \tag{52}$$

$$= \overline{P}^+\left(s_{t+1} \middle| s_t, b_t^{\kappa,\mu}, a_t\right) P^+\left(\tilde{b}_{t+1}^{\kappa,\mu} \middle| s_t, b_t^{\kappa,\mu}, c_t\right) \tag{53}$$

$$= \mathbb{E}_{(\kappa_i,\mu_i)\sim b_t^{\kappa,\mu}}\left[P^{\kappa_i}(s_{t+1}|s_t,a_t)\right] \cdot \mathbb{1}\left[\tilde{b}_{t+1}^{\kappa,\mu} = p(b_t^{\kappa,\mu}|s_t,c_t)\right], \tag{54}$$

$\overline{P}_0^+ : \Delta\left(\overline{\mathcal{S}}^+\right)$ is the initial hyper-state distribution, i.e., a deterministic initial hyper-state is

$$\bar{s}_0^+ = (s_0, b_0^{\kappa,\mu}) = (s_0, p(\kappa,\mu)) \in \overline{\mathcal{S}}^+, \tag{55}$$

and $\overline{R}^+ : \overline{\mathcal{S}}^+ \times \mathcal{A} \to \Delta(\mathcal{R})$ is the reward distribution: $\forall \bar{s}^+ = (s, b^{\kappa,\mu}) \in \overline{\mathcal{S}}^+, a \in \mathcal{A}, r \in \mathcal{R}$,

$$\overline{R}^+\left(r|\bar{s}^+, a\right) = \overline{R}^+\left(r \middle| s, b^{\kappa,\mu}, a\right) = \mathbb{E}_{(\kappa_i,\mu_i)\sim b^{\kappa,\mu}}\left[R^{\kappa_i}(r|s,a)\right]. \tag{56}$$

In a transformed BAMDP $\overline{M}^+$, the overall belief is about the task-dependent behavior policies, transition function, and reward function, which are constant for a given task. A meta-policy on $\overline{M}^+$ is $\bar{\pi}^+ : \overline{\mathcal{S}}^+ \to \Delta(\mathcal{A})$ prescribes a distribution over actions for each hyper-state. With feasible Bayesian belief updating, the objective of RL agents is now to find a meta-policy on hyper-states $\bar{\pi}^+$ that maximizes the expected return in the transformed BAMDP,

$$\mathcal{J}_{\overline{M}^+}\left(\bar{\pi}^+\right) = \mathbb{E}_{a_t\sim\bar{\pi}^+\left(\cdot\middle|\bar{s}_t^+\right), r_t\sim\overline{R}^+\left(\cdot\middle|\bar{s}_t^+, a_t\right), s_{t+1}^+\sim\overline{P}^+\left(\cdot\middle|\bar{s}_t^+, a_t\right)}\left[\sum_{t=0}^{\widehat{H}-1} r_t\right] \tag{57}$$

$$= \mathbb{E}_{(\kappa_i,\mu_i)\sim p(\kappa,\mu)}\left[\sum_{j=0}^{N-1} \mathbb{E}_{a_t\sim\bar{\pi}^+\left(\cdot\middle|\bar{s}_{jH+t}^+\right), r_t\sim R^{\kappa_i}(\cdot|s_t,a_t), s_{t+1}\sim P^{\kappa_i}(\cdot|s_t,a_t)}\left[\sum_{t=0}^{H-1} r_t\right]\right] \tag{58}$$

$$= \mathbb{E}_{\kappa_i\sim p(\kappa)}\left[\sum_{j=0}^{N-1} \mathbb{E}_{a_t\sim\bar{\pi}^+\left(\cdot\middle|\bar{s}_{jH+t}^+\right), r_t\sim R^{\kappa_i}(\cdot|s_t,a_t), s_{t+1}\sim P^{\kappa_i}(\cdot|s_t,a_t)}\left[\sum_{t=0}^{H-1} r_t\right]\right]. \tag{59}$$

For any meta-policy on hyper-states $\bar{\pi}^+$, denote the corresponding meta-policy on history contexts $\hat{f}_{\bar{\pi}^+} : \mathcal{C} \to \Delta(\mathcal{A})$, i.e., $\forall t \in \left[\widehat{H}-1\right], \forall c_{:t} \in \mathcal{C}_t$, s.t., $\hat{f}_{\bar{\pi}^+}(\cdot|c_{:t}) = \bar{\pi}^+\left(\cdot\middle|\bar{s}_t^+\right)$, where $\bar{s}_t^+ = (s_t, b_t^{\kappa,\mu}) = (s_t, p(\kappa,\mu|c_{:t}))$, and we have

$$\mathcal{J}_{\widehat{M}}\left(\hat{f}_{\bar{\pi}^+}\right) = \mathbb{E}_{\kappa_i\sim p(\kappa)}\left[\sum_{j=0}^{N-1} \mathbb{E}_{a_t\sim\hat{f}_{\bar{\pi}^+}\left(\cdot\middle|c_{:(jH+t)}\right), r_t\sim R^{\kappa_i}(\cdot|s_t,a_t), s_{t+1}\sim P^{\kappa_i}(\cdot|s_t,a_t)}\left[\sum_{t=0}^{H-1} r_t\right]\right] \tag{60}$$

$$= \mathcal{J}_{\overline{M}^+}\left(\bar{\pi}^+\right). \tag{61}$$

Lemma 1. *In an MDP M, for each behavior policy μ and batch-constrained policy π, collect a dataset $\mathcal{D}$ and the gap between approximate offline policy evaluation $\mathcal{J}_{\mathcal{D}}(\pi)$ and accurate policy evaluation $\mathcal{J}_M(\pi)$ will asymptotically approach to 0, as the offline dataset $\mathcal{D}$ grows.*

From a given dataset $\mathcal{D}$, an abstract MDP $M_{\mathcal{D}}$ can be estimated (Fujimoto et al., 2019; Yin & Wang, 2021; Szepesvári, 2022). According to concentration bounds, the estimated transition and reward function will asymptotically approach M (Yin & Wang, 2021) during the support of $\mathcal{D}$. Then, using the simulation lemma (Alekh Agarwal, 2017; Szepesvári, 2022), the gap between $\mathcal{J}_{\mathcal{D}}(\pi)$ and $\mathcal{J}_M(\pi)$ will asymptotically approach to 0, as the offline dataset $\mathcal{D}$ grows. Formal proofs are deferred in Appendix A.6.

Theorem 2. *In a BAMDP M^+, for each task-dependent behavior policy $[\mu]$, denoting a transformed BAMDP $\overline{M}^+$ by incorporating $[\mu]$ into the belief of M^+, we have (i) for feasible Bayesian belief updating, $\overline{M}^+$ confines the agent in the in-distribution hyper-states, (ii) for each $\bar{\pi}^+$, the distribution of $\mathbb{P}_{\overline{M}^+,[\mu]}$ and $\mathbb{P}_{\overline{M}^+,\bar{\pi}^+}$ matches, and (iii) policy evaluation $\mathcal{J}_{\overline{\mathcal{D}}^+}(\bar{\pi}^+)$ and $\mathcal{J}_{\overline{M}^+}(\bar{\pi}^+)$ will be asymptotically consistent, as the offline dataset grows.*

Proof. **Part (i)** During online adaptation, RL agents construct a hyper-state $\bar{s}_t^+ = \left(s_t, \bar{b}_t\right)$ from the context history and perform a meta-policy $\bar{\pi}^+\left(a_t \left| \bar{s}_t^+\right.\right)$. The new belief $\bar{b}_t$ accounts for the uncertainty of task MDPs and task-dependent behavior policies. In contrast with Proposition 1(i), for feasible Bayesian belief updating, transformed BAMDPs do not allow the agent to visit out-of-distribution hyper-states. Otherwise, the context history will conflict with the belief about behavior policies, i.e., RL agents cannot update their beliefs $\bar{b}_t$ when they have observed an event that they believe to have probability zero.

Part (ii) We assume feasible Bayesian belief updating in this proof. At first, $\forall s_t^+, a_t$, s.t. $p^{\bar{\pi}^+}_{\overline{M}^+}\left(\bar{s}_t^+, a_t\right) > 0$, we aim to prove

$$\mathbb{P}_{\overline{M}^+,[\mu]}\left(r_t, s_{t+1} \left| \bar{s}_t^+, a_t\right.\right) = \mathbb{P}_{\overline{M}^+,\bar{\pi}^+}\left(r_t, s_{t+1} \left| \bar{s}_t^+, a_t\right.\right). \tag{62}$$

Since $\bar{\pi}^+$ is batch-constrained policy by $[\mu]$, if $p^{\bar{\pi}^+}_{\overline{M}^+}\left(\bar{s}_t^+, a_t\right) > 0$, we have $p^{[\mu]}_{\overline{M}^+}\left(\bar{s}_t^+, a_t\right) > 0$. Then,

$$\begin{aligned}
\mathbb{P}_{\overline{M}^+,\bar{\pi}^+}\left(r_t, s_{t+1} \left| \bar{s}_t^+, a_t\right.\right) &= \overline{R}^+\left(r_t \left| \bar{s}_t^+, a_t\right.\right) \cdot \overline{P}^+\left(s_{t+1} \left| \bar{s}_t^+, a_t\right.\right) && (63)\\
&= \mathbb{E}_{\kappa_i \sim b_t^{\kappa,\mu}}\left[\mathbb{P}_{\kappa_i}\left(r_t, s_{t+1} \left| s_t, a_t\right.\right)\right] && (64)\\
&= \mathbb{E}_{(\kappa_i,\mu_i)\sim b_t^{\kappa,\mu}}\left[\mathbb{P}_{\kappa_i}\left(r_t, s_{t+1} \left| s_t, a_t\right.\right)\right] && (65)
\end{aligned}$$

where $b_t^{\kappa,\mu}$ is defined in Eq. (48) and

$$p\left(\kappa_i, \mu_i \left| b_t^{\kappa,\mu}\right.\right) = \mathbb{E}_{\kappa_i\sim p(\kappa),\mu_i\sim p(\mu|\kappa_i)}\left[\mathbb{E}_{c_{:t}\sim p_{\overline{M}^+}(c_{:t}|\kappa_i,\mu_i)}\left[\mathbb{1}\left[b_t^{\kappa,\mu} = p(\kappa,\mu|c_{:t})\right]\right]\right], \tag{66}$$

where $p_{\overline{M}^+}\left(c_{:t}|\kappa_i,\mu_i\right)$ is defined in Eq. (6). According to Eq. (37) and (38),

$$\begin{aligned}
&\mathbb{P}_{\overline{M}^+,[\mu]}\left(r_t, s_{t+1} \left| \bar{s}_t^+, a_t\right.\right) && (67)\\
\propto\ &\mathbb{E}_{\kappa_i\sim p(\kappa),\mu_i\sim p(\mu|\kappa_i)}\left[\mathbb{P}_{\kappa_i}\left(r_t, s_{t+1} \left| s_t, a_t\right.\right) \cdot p_{\overline{M}^+}\left(\bar{s}_t^+ \left| \kappa_i, \mu_i\right.\right)\right] && (68)\\
=\ &\mathbb{E}_{\kappa_i\sim p(\kappa),\mu_i\sim p(\mu|\kappa_i)}\left[\mathbb{E}_{c_{:t}\sim p_{\overline{M}^+}(c_{:t}|\kappa_i,\mu_i)}\left[\mathbb{P}_{\kappa_i}\left(r_t, s_{t+1} \left| s_t, a_t\right.\right) \cdot \mathbb{1}\left[b_t^{\kappa,\mu} = p(\kappa,\mu|c_{:t})\right]\right]\right] && (69)\\
=\ &\mathbb{E}_{(\kappa_i,\mu_i)\sim b_t^{\kappa,\mu}}\left[\mathbb{P}_{\kappa_i}\left(r_t, s_{t+1} \left| s_t, a_t\right.\right)\right] && (70)\\
=\ &\mathbb{P}_{\overline{M}^+,\bar{\pi}^+}\left(r_t, s_{t+1} \left| \bar{s}_t^+, a_t\right.\right). && (71)
\end{aligned}$$

Thus, the data distribution induced by $\bar{\pi}^+$ and $[\mu]$ matches.

Part (iii) Directly use Lemma 1 in a transformed BAMDP $\overline{M}^+$, in which $\overline{M}^+$ is a belief MDP, a type of MDP. Therefore, the policy evaluation of $\bar{\pi}^+$ in offline meta-training and online adaptation will be asymptotically consistent, as the offline dataset grows. □

To achieve in-distribution online adaptation, transformed BAMDPs incorporate additional information about offline data collection into the beliefs of BAMDPs. We prove that transformed BAMDPs require RL agents to filter out out-of-distribution episodes to support feasible belief updating of behavior policies. In this way, the distribution of reward and transition between offline and online data collection coincide, which can provide the guarantee of consistent policy evaluation between $\mathcal{J}_{\overline{\mathcal{D}}^+}(\bar{\pi}^+)$ and $\mathcal{J}_{\overline{M}^+}(\bar{\pi}^+)$. Theorem 2 shows that we can meta-train policies with offline policy evaluation and utilize in-distribution online adaptation to guarantee the final performance in meta-testing.

A.4. Main Results in Section 3.3

Definition 3 (Sub-Datasets Collected by Single Task Data Collection)**.** *In a transformed BAMDP $\overline{M}^+$, an offline multi-task dataset $\overline{\mathcal{D}}^+$ is drawn from the task-dependent data collection $\mathbb{P}_{\overline{M}^+,[\mu]}$. A sub-dataset collected by a behavior policy μ_i in a*

task κ_i is defined by $\mathcal{D}_{\kappa_i,\mu_i}$. Note an offline multi-task dataset $\overline{\mathcal{D}}^+$ is the union of sub-datasets $\mathcal{D}_{\kappa_i,\mu_i}$, i.e.,

$$\overline{\mathcal{D}}^+ = \bigcup_{\kappa_i,\mu_i} \mathcal{D}_{\kappa_i,\mu_i}. \tag{72}$$

For each sub-dataset $\mathcal{D}_{\kappa_i,\mu_i}$, we can define a batch-constrained policy set in a single-task (κ_i, μ_i) as $\Pi^{\mathcal{D}_{\kappa_i,\mu_i}}$ (see the definition in Eq. (12)).

Definition 4 (Meta-Policy with Thompson Sampling)**.** *For each transformed BAMDP $\overline{M}^+$, a meta-policy set with Thompson sampling on $\overline{M}^+$ is defined by $\bar{\pi}^{+,T} : \mathcal{S} \times \overline{\mathcal{B}} \times \mathcal{K} \times \Pi_{[\mu]} \to \Delta(\mathcal{A})$, where $\overline{\mathcal{B}}$ is the space of beliefs over meta-RL MDPs with behavior policies, $\mathcal{K}$ is the space of task κ, and $\Pi_{[\mu]}$ is the space of task-dependent behavior policies. In each episode, $\bar{\pi}^{+,T}$ samples a task hypothesis (κ_i, μ_i) from the current belief $b_{t'}^{\kappa,\mu}$, where t' is the starting step in this episode. During this episode, $\bar{\pi}^{+,T}(\cdot \,|s_t, b_{t'}^{\kappa,\mu}, \kappa_i, \mu_i)$ prescribes a distribution over actions for each state s_t, belief $b_{t'}^{\kappa,\mu}$, and task hypothesis (κ_i, μ_i). Beliefs $b_{t'}^{\kappa,\mu}$ and task hypotheses (κ_i, μ_i) will periodically update after each episode.*

In the deep-learning-based implementation, a context-based meta-RL algorithm, PEARL (Rakelly et al., 2019), utilizes a meta-policy with Thompson sampling (Strens, 2000) to iteratively update task belief by interacting with the environment and improve the meta-policy based on the "task hypothesis" sampled from the current beliefs. We can adopt such adaptation protocol to design practical offline meta-RL algorithms for transformed BAMDPs.

Definition 5 (Batch-Constrained Meta-Policy Set with Thompson Sampling)**.** *For each transformed BAMDP $\overline{M}^+$ with an offline multi-task dataset $\overline{\mathcal{D}}^+$, a batch-constrained meta-policy set with Thompson sampling is defined by*

$$\Pi^{\overline{\mathcal{D}}^+,T} = \left\{ \bar{\pi}^{+,T} \,\middle|\, \bar{\pi}^{+,T}(a_t \,|s_t, b_{t'}^{\kappa,\mu}, \kappa_i, \mu_i) = 0 \text{ whenever } (s_t, a_t) \notin \mathcal{D}_{\kappa_i,\mu_i}, \forall b_{t'}^{\kappa,\mu} \right\}, \tag{73}$$

where denoting $(s_t, a_t) \in \mathcal{D}_{\kappa_i,\mu_i}$ if there exists a trajectory containing (s_t, a_t) in the dataset $\mathcal{D}_{\kappa_i,\mu_i}$.

The batch-constrained meta-policy set with Thompson sampling $\Pi^{\overline{\mathcal{D}}^+,T}$ consists of the meta-policies that for any state s_t observed in the hypothesis dataset $\mathcal{D}_{\kappa_i,\mu_i}$, the agent will not select an action outside of the dataset. Note that in each episode with a task hypothesis (κ_i, μ_i), a batch-constrained meta-policy with Thompson sampling $\bar{\pi}^{+,T}$ is batch-constrained within a sub-dataset $\mathcal{D}_{\kappa_i,\mu_i}$, i.e., $\forall b_{t'}^{\kappa,\mu}$, we have $\bar{\pi}^{+,T}(\cdot \,|s_t, b_{t'}^{\kappa,\mu}, \kappa_i, \mu_i) \in \Pi^{\mathcal{D}_{\kappa_i,\mu_i}}$.

Definition 6 (Probability that a Policy Leaves the Dataset)**.** *In an MDP M and an arbitrary offline dataset $\mathcal{D}$, for each policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$, the probability that executing π in M leaves the dataset $\mathcal{D}$ for an episode is defined by*

$$p_{out}^{M,\mathcal{D}}(\pi) = \sum_{\tau_H} p_\pi^M(\tau_H)\, \mathbb{1}\left[\tau_H \text{ leaves } \mathcal{D}\right] \tag{74}$$

$$= \sum_{\tau_H} p_\pi^M(\tau_H)\, \mathbb{1}\left[\exists t \in [H] \text{ s.t. } s_t \notin \mathcal{D} \text{ or } (s_t, a_t, r_t) \notin \mathcal{D}\right], \tag{75}$$

where $p_\pi^M(\tau_H)$ is the probability of executing π in M to generate an H-horizon trajectory τ_H (see the definition in Eq. (6)), denoting $s_t \in \mathcal{D}$ if there exists a trajectory containing s_t in the dataset $\mathcal{D}$, and similarly for $(s_t, a_t, r_t) \in \mathcal{D}$.

When we aim to confine the agent in the in-distribution states with high probability as the offline dataset $\mathcal{D}$ grows, it is equivalent to binding the probability that executing a policy π in M leaves the dataset $\mathcal{D}$ for an episode, i.e., $p_{out}^{M,\mathcal{D}}(\pi)$.

Theorem 3. *In a transformed BAMDP $\overline{M}^+$, for each batch-constrained meta-policy with Thompson sampling $\bar{\pi}^{+,T}$ in a meta-testing task $\kappa_{test} \sim p(\kappa)$, there exists a task hypothesis from the current belief, executing $\bar{\pi}^{+,T}$ in κ_{test} can generate in-distribution online adaptation episodes with high probability, as the offline dataset grows.*

Proof. Denote the current belief by $b_{t'}^{\kappa,\mu}$ and the task hypothesis by (κ_i, μ_i). Thus, for each batch-constrained meta-policy with Thompson sampling $\bar{\pi}^{+,T} \in \Pi^{\overline{\mathcal{D}}^+,T}$ with $(b_{t'}^{\kappa,\mu}, \kappa_i, \mu_i)$, similar to Definition 6, define the probability that executing $\bar{\pi}^{+,T}(\cdot \,|s_t, b_{t'}^{\kappa,\mu}, \kappa_i, \mu_i)$ leaves the dataset $\overline{\mathcal{D}}^+$ in an adaptation episode of a meta-testing task $\kappa_{test} \sim p(\kappa)$:

$$p_{out}^{\kappa_{test},\overline{\mathcal{D}}^+}\left(\bar{\pi}^{+,T}, b_{t'}^{\kappa,\mu}, \kappa_i, \mu_i\right) = \sum_{\bar{\tau}_H^+} p_{\bar{\pi}^{+,T}}^{\kappa_{test}}\left(\bar{\tau}_H^+ \middle| b_{t'}^{\kappa,\mu}, \kappa_i, \mu_i\right) \mathbb{1}\left[\bar{\tau}_H^+ \text{ leaves } \overline{\mathcal{D}}^+\right], \tag{76}$$

where $p^{\kappa_{test}}_{\bar{\pi}^{+,T}}\left(\bar{\tau}^{+}_{H}\middle|\, b^{\kappa,\mu}_{t'}, \kappa_i, \mu_i\right)$ is the probability of executing $\bar{\pi}^{+,T}\left(\cdot\,|s_t, b^{\kappa,\mu}_{t'}, \kappa_i, \mu_i\,\right)$ in κ_{test} to generate an H-horizon trajectory $\bar{\tau}^{+}_{H}$ in an adaptation episode, i.e.,

$$
\begin{aligned}
& p^{\kappa_{test}}_{\bar{\pi}^{+,T}}\left(\bar{\tau}^{+}_{H}\middle|\, b^{\kappa,\mu}_{t'}, \kappa_i, \mu_i\right) && (77)\\
&= \left(\prod_{t=t'}^{t'+H-1} \bar{\pi}^{+,T}\left(a_t\,|s_t, b^{\kappa,\mu}_{t'}, \kappa_i, \mu_i\,\right) \cdot R^{\kappa_{test}}\left(r_t|s_t, a_t\right)\right) \prod_{t=t'}^{t'+H-2} P^{\kappa_{test}}\left(s_{t+1}\,|s_t, a_t, r_t\,\right) && (78)\\
&= p^{\kappa_{test}}_{\bar{\pi}^{+,T}}\left(\tau_H\middle|\, b^{\kappa,\mu}_{t'}, \kappa_i, \mu_i\right), && (79)
\end{aligned}
$$

where we can transform $\bar{\tau}^{+}_{H}$ to τ_H with the same probability since the belief $b^{\kappa,\mu}_{t'}$ and task hypothsis (κ_i, μ_i) will periodically update after each episode. Therefore,

$$
\begin{aligned}
p^{\kappa_{test},\overline{\mathcal{D}}^{+}}_{out}\left(\bar{\pi}^{+,T}, b^{\kappa,\mu}_{t'}, \kappa_i, \mu_i\right) &= \sum_{\bar{\tau}^{+}_{H}} p^{\kappa_{test}}_{\bar{\pi}^{+,T}}\left(\tau_H\middle|\, b^{\kappa,\mu}_{t'}, \kappa_i, \mu_i\right) \mathbb{1}\left[\bar{\tau}^{+}_{H} \text{ leaves } \overline{\mathcal{D}}^{+}\right] && (80)\\
&\leq \sum_{\tau_H} p^{\kappa_{test}}_{\bar{\pi}^{+,T}}\left(\tau_H\middle|\, b^{\kappa,\mu}_{t'}, \kappa_i, \mu_i\right) \mathbb{1}\left[\tau_H \text{ leaves } \mathcal{D}_{\kappa_{i^*},\mu_{i^*}}\right] && (81)\\
&= p^{\kappa_{test},\mathcal{D}_{\kappa_{i^*},\mu_{i^*}}}_{out}\left(\bar{\pi}^{+,T}\middle|\, b^{\kappa,\mu}_{t'}, \kappa_i, \mu_i\right), && (82)
\end{aligned}
$$

where $p^{\kappa_{test},\mathcal{D}_{\kappa_{i^*},\mu_{i^*}}}_{out}\left(\bar{\pi}^{+,T}\middle|\, b^{\kappa,\mu}_{t'}, \kappa_i, \mu_i\right)$ is the probability that executing $\bar{\pi}^{+,T}\left(\cdot\,|s_t, b^{\kappa,\mu}_{t'}, \kappa_i, \mu_i\,\right)$ in κ_{test} leaves the sub-dataset $\mathcal{D}_{\kappa_{i^*},\mu_{i^*}}$ for an episode (see Definition 6), $\mathcal{D}_{\kappa_{i^*},\mu_{i^*}}$ is a sub-dataset collected in $\overline{\mathcal{D}}^{+}$ (see Definition 3) and κ_{i^*} is the closest offline meta-training task to κ_{test}, i.e.,

$$
\begin{aligned}
\kappa_{i^*} &= \operatorname*{arg\,min}_{\kappa_i \in \mathcal{K}_{train}} \left\|\kappa_i - \kappa_{test}\right\|_{\infty} && (83)\\
&= \operatorname*{arg\,min}_{\kappa_i \in \mathcal{K}_{train}} \max\left(\left\|P^{\kappa_i}(s,a,s') - P^{\kappa_{test}}(s,a,s')\right\|_{\infty}, \left\|R^{\kappa_i}(s,a,r) - R^{\kappa_{test}}(s,a,r)\right\|_{\infty}\right), && (84)
\end{aligned}
$$

in which denoting the i.i.d. offline meta-training tasks sampled from $p(\kappa)$ in $\overline{\mathcal{D}}^{+}$ by $\mathcal{K}_{train}$. From Lemma 5, as the offline dataset $\overline{\mathcal{D}}^{+}$ grows, $\mathcal{D}_{\kappa_{i^*},\mu_{i^*}}$ and $\mathcal{K}_{train}$ grow monotonically, for any batch-constrained policy π in $\mathcal{D}_{\kappa_{i^*},\mu_{i^*}}$, i.e., $\forall \pi \in \Pi^{\mathcal{D}_{\kappa_{i^*},\mu_{i^*}}}$, when executing π in an episode of $\kappa_{test} \sim p(\kappa)$, the probability leaving the dataset $\overline{\mathcal{D}}^{+}$ is $p^{\kappa_{test},\mathcal{D}_{\kappa_{i^*},\mu_{i^*}}}_{out}(\pi)$, which asymptotically approaches zero.

For the first adaptation episode in a meta-testing task $\kappa_{test} \sim p(\kappa)$ with the prior belief $p(\kappa,\mu) = p(\kappa)p(\mu|\kappa)$, there exists a task hypothesis $(\kappa_{i^*}, \mu_{i^*})$ in the prior $p(\kappa,\mu)$, then due to $\bar{\pi}^{+,T}\left(\cdot\,|s_t, p(\kappa,\mu), \kappa_{i^*}, \mu_{i^*}\,\right) \in \Pi^{\mathcal{D}_{\kappa_{i^*},\mu_{i^*}}}$ from Definition 5 and

$$
p^{\kappa_{test},\overline{\mathcal{D}}^{+}}_{out}\left(\bar{\pi}^{+,T}, p(\kappa,\mu), \kappa_{i^*}, \mu_{i^*}\right) \leq p^{\kappa_{test},\mathcal{D}_{\kappa_{i^*},\mu_{i^*}}}_{out}\left(\bar{\pi}^{+,T}\middle|\, p(\kappa,\mu), \kappa_{i^*}, \mu_{i^*}\right), \tag{85}
$$

as the offline dataset $\overline{\mathcal{D}}^{+}$ grows, executing $\bar{\pi}^{+,T}$ with $(p(\kappa,\mu), \kappa_{i^*}, \mu_{i^*})$ for the first episode in κ_{test} will confine the agent in in-distribution hyper-states with high probability. In the subsequent adaptation episodes with current belief $b^{\kappa,\mu}_{t'}$ in κ_{test}, the task hypothesis $(\kappa_{i^*}, \mu_{i^*})$ is also in the belief $b^{\kappa,\mu}_{t'}$ by induction.

Therefore, for each adaptation episode with current belief $b^{\kappa,\mu}_{t'}$ in κ_{test}, there exists a task hypothesis (κ_i, μ_i) from $b^{\kappa,\mu}_{t'}$, e.g., $(\kappa_{i^*}, \mu_{i^*})$, executing $\bar{\pi}^{+,T}$ with $(b^{\kappa,\mu}_{t'}, \kappa_i, \mu_i)$ in κ_{test} will confine the agent in in-distribution hyper-states with high probability, as the offline dataset $\overline{\mathcal{D}}^{+}$ grows. □

Theorem 3 indicates that for each adaptation episode, we can sample task hypotheses from the current task belief and execute $\bar{\pi}^{+,T}$ to interact with the environment until finding an in-distribution episode. For example in Figure 2, after offline meta-training, a meta-policy with Thompson sampling $\bar{\pi}^{+,T}$ will perform a_i with a task hypothesis of κ_i and expect to receive a reward 1. During online meta-testing, κ_{test} is drawn from $p(\kappa)$ and the agent needs to infer the task identification. To achieve in-distribution online adaptation, $\bar{\pi}^{+,T}$ will try various actions according to diverse task hypotheses until sampling an in-distribution episode with a reward 1. Updating the task belief with the in-distribution episode, RL agents can infer and solve this task.

In contrast, when updating task belief using an out-of-distribution episode with a reward 0, the posterior task belief will be out of the offline dataset $\overline{\mathcal{D}}^+$. Note that offline training paradigm can not well-optimize $\bar{\pi}^{+,T}$ on out-of-distribution states (Fujimoto et al., 2019) and policy $\bar{\pi}^{+,T}$ will fail in this case. Moreover, Thompson sampling is very popular in context-based deep meta-RL (Rakelly et al., 2019) and we will generalize these theoretical implications.

Note that Theorem 3 considers arbitrary task distribution $p(\kappa)$, since the distance between the closest meta-training task κ_{i^*} and κ_{test} will asymptotically approach zero with high probability, as the i.i.d. offline meta-training tasks $\mathcal{K}_{train}$ sampled from $p(\kappa)$ in $\overline{\mathcal{D}}^+$ grows.

A.5. Omitted Assumptions and Propsitions in Section 4

To analyze return-based uncertainty quantification, we first present a mild assumption:

Assumption 1. *In a transformed BAMDP $\overline{M}^+$, after effective offline meta-training of meta-policy with Thompson sampling $\bar{\pi}^{+,T}$, with high probability, there exists (accurate or nearly accurate) task hypotheses from the current belief, the online policy evaluation has higher returns on in-distribution episodes than out-of-distribution episodes with incorrect hypotheses.*

This assumption matches the bias of offline RL (Fujimoto et al., 2019) that offline meta-training can not well-optimize meta-policies on out-of-distribution states.

Proposition 2. *In a transformed BAMDP $\overline{M}^+$, after effective offline meta-training of meta-policy with Thompson sampling $\bar{\pi}^{+,T}$, with Assumption 1, return-based uncertainty quantification $\mathbb{Q}_{RE}$ can support $\bar{\pi}^{+,T}$ generate in-distribution online adaptation episodes.*

Proof. From Theorem 3, meta-policy with Thompson sampling $\bar{\pi}^{+,T}$ and accurate or nearly accurate task hypotheses can generate in-distribution online episodes with high probability. Combining with Assumption 1, for each adaptation episode, there exists (accurate or nearly accurate) task hypotheses whose online policy evaluation has higher returns. Thus, from the current belief, we can sample task hypotheses and execute $\bar{\pi}^{+,T}$ to interact with the environment until finding an in-distribution episode using $\mathbb{Q}_{RE}$. □

A.6. Omitted Proof of Lemma 1

Definition 7 (Dataset Induced Finite-Horizon MDPs)**.** *In a finite-horizon MDP M with an offline dataset $\mathcal{D}$, a dataset induced finite-horizon MDP is defined by $M^{\mathcal{D}} = \left(\mathcal{S}, \mathcal{A}, \mathcal{R}, H, P^{M^{\mathcal{D}}}, R^{M^{\mathcal{D}}}\right)$, with the same state space, action space, reward space, and horizon as M. The transition function is defined as follows: $\forall s, s' \in \mathcal{S}, a \in \mathcal{A}$,*

$$P^{M^{\mathcal{D}}}(s'|s,a) = \begin{cases} \dfrac{N(s,a,s')}{N(s,a)}, & \text{if } N(s,a) > 0, \\ 0, & \text{otherwise}, \end{cases} \tag{86}$$

where $N(s,a,s')$ and $N(s,a)$ are the number of times the tuples (s,a,s') and (s,a) are observed in $\mathcal{D}$, respectively. The reward function is defined by $\forall s \in \mathcal{S}, a \in \mathcal{A}, r \in \mathcal{R}$,

$$R^{M^{\mathcal{D}}}(r|s,a) = \begin{cases} \dfrac{N(s,a,r)}{N(s,a)}, & \text{if } N(s,a) > 0, \\ 0, & \text{otherwise}, \end{cases} \tag{87}$$

where $N(s,a,r)$ is the number of times the tuple (s,a,r) are observed in $\mathcal{D}$. The offline policy evaluation in $\mathcal{D}$ is equal to the policy evaluation in $M_{\mathcal{D}}$, i.e., for any batch-constrained policy π,

$$\mathcal{J}_{\mathcal{D}}(\pi) = \mathcal{J}_{M_{\mathcal{D}}}(\pi). \tag{88}$$

Note that dataset induced finite-horizon MDPs $M^{\mathcal{D}}$ are not defined on supports outside of dataset $\mathcal{D}$. For simplicity, We set all undefined numbers to 0 in the transition and reward function.

Lemma 2 (Simulation Lemma for Offline Finite-Horizon MDPs)**.** *In an MDP M with an offline dataset $\mathcal{D}$, for any batch-constrained policy $\pi \in \Pi^{\mathcal{D}}$, if*

$$\max_{s\in\mathcal{S}, a\in\mathcal{A} \text{ with } \rho_\pi^{M_{\mathcal{D}}}(s,a)>0} \left\|P^{M_{\mathcal{D}}}(\cdot|s,a) - P^{M}(\cdot|s,a)\right\|_1 \le \epsilon_P, \tag{89}$$

$$\max_{s\in\mathcal{S}, a\in\mathcal{A} \text{ with } \rho_\pi^{M_{\mathcal{D}}}(s,a)>0} \left|r^{M_{\mathcal{D}}}(s,a) - r^{M}(s,a)\right| \le \epsilon_r, \tag{90}$$

$$\max_{s\in\mathcal{S}, a\in\mathcal{A}} \max\left(r^{M_{\mathcal{D}}}(s,a), r^{M}(s,a)\right) \le r_{max}, \tag{91}$$

where $r^M(s,a) = \mathbb{E}_{\tilde{r}\sim R^M(s,a)}[\tilde{r}]$ and $r^{M_{\mathcal{D}}}(s,a) = \mathbb{E}_{\tilde{r}\sim R^{M_{\mathcal{D}}}(s,a)}[\tilde{r}]$, we have

$$|\mathcal{J}_{M_{\mathcal{D}}}(\pi) - \mathcal{J}_M(\pi)| \le H\epsilon_r + \frac{H(H-1)r_{max}}{2}\epsilon_P. \tag{92}$$

Proof. Similar to the famous Simulation Lemma in finite-horizon MDPs (Alekh Agarwal, 2017), the proof is as follows. Recall value function $\forall h \in [H-1], \forall s \in \mathcal{S}_h$ (see Eq. (7)),

$$V_\pi^M(s) = \sum_{a\in\mathcal{A}} \pi(a|s) \left(r^M(s,a) + \sum_{s'\in\bar{\mathcal{S}}_{h+1}} P^M(s'|s,a) V_\pi^M(s') \right), \tag{93}$$

$\mathcal{J}_M(\pi) = V_\pi^M(s_0)$, and $\forall h \in [H]$,

$$\max_{s\in\bar{\mathcal{S}}_h} \left|V_\pi^{\bar{\kappa}}(s)\right| \le (H-h)r_{max}. \tag{94}$$

We will prove $\forall h \in [H], \forall s \in \mathcal{S}_h$ with $\rho_\pi^{M_{\mathcal{D}}}(s) > 0$,

$$\left|V_\pi^{M_{\mathcal{D}}}(s) - V_\pi^M(s)\right| \le (H-h)\epsilon_r + \frac{(H-h)(H-h-1)r_{max}}{2}\epsilon_P \tag{95}$$

by induction. When $h = H-1$, we have $\forall s \in \mathcal{S}_h$ with $\rho_\pi^{M_\mathcal{D}}(s) > 0$,

$$\left|V_\pi^{M_\mathcal{D}}(s) - V_\pi^M(s)\right| = \left|\sum_{a\in\mathcal{A}} \pi(a|s) r^{M_\mathcal{D}}(s,a) - \sum_{a\in\mathcal{A}} \pi(a|s) r^M(s,a)\right| \le \epsilon_r \tag{96}$$

holds. And $\forall h \in [H-1], \forall s \in \mathcal{S}_h$ with $\rho_\pi^{M_\mathcal{D}}(s) > 0$,

$$\begin{aligned}
&\left|V_\pi^{M_\mathcal{D}}(s) - V_\pi^M(s)\right| && (97)\\
&= \left|\sum_{a\in\mathcal{A}} \pi(a|s) \left(r^{M_\mathcal{D}}(s,a) + \sum_{s'\in\mathcal{S}_{h+1}} P^{M_\mathcal{D}}(s'|s,a) V_\pi^{M_\mathcal{D}}(s')\right) - \right. && (98)\\
&\qquad \left. \sum_{a\in\mathcal{A}} \pi(a|s) \left(r^M(s,a) + \sum_{s'\in\mathcal{S}_{h+1}} P^M(s'|s,a) V_\pi^M(s')\right)\right| && (99)\\
&\le \sum_{a\in\mathcal{A}} \pi(a|s) \left|r^{M_\mathcal{D}}(s,a) - r^M(s,a)\right| + && (100)\\
&\quad \sum_{a\in\mathcal{A}} \pi(a|s) \sum_{s'\in\mathcal{S}_{h+1}} \left|P^{M_\mathcal{D}}(s'|s,a) V_\pi^{M_\mathcal{D}}(s') - P^M(s'|s,a) V_\pi^M(s')\right| && (101)\\
&\le \epsilon_r + \sum_{a\in\mathcal{A}} \pi(a|s) \sum_{s'\in\mathcal{S}_{h+1}} \left|P^{M_\mathcal{D}}(s'|s,a) - P^M(s'|s,a)\right| V_\pi^M(s') + && (102)\\
&\quad \sum_{a\in\mathcal{A}} \pi(a|s) \sum_{s'\in\mathcal{S}_{h+1}} P^{M_\mathcal{D}}(s'|s,a) \left|V_\pi^{M_\mathcal{D}}(s') - V_\pi^M(s')\right| && (103)\\
&\le \epsilon_r + (H-(h+1)) r_{max} \epsilon_P + && (104)\\
&\quad \left((H-(h+1))\epsilon_r + \frac{(H-(h+1))(H-(h+1)-1) r_{max}}{2} \epsilon_P\right) && (105)\\
&= (H-h)\epsilon_r + \frac{(H-h)(H-h-1) r_{max}}{2} \epsilon_P. && (106)
\end{aligned}$$

Thus,

$$\left|\mathcal{J}_{M_\mathcal{D}}(\pi) - \mathcal{J}_M(\pi)\right| = \left|V_\pi^{M_\mathcal{D}}(s_0) - V_\pi^M(s_0)\right| \le H\epsilon_r + \frac{H(H-1) r_{max}}{2} \epsilon_P. \tag{107}$$

□

Lemma 3. *In an MDP M with an offline dataset $\mathcal{D}$ collected by a behavior policy μ, for any batch-constrained policy $\pi \in \Pi^\mathcal{D}$, $\forall \delta \in (0,1]$, with probability $1-\delta$,*

$$\left|\mathcal{J}_{M_\mathcal{D}}(\pi) - \mathcal{J}_M(\pi)\right| \le H^2 \left|\mathcal{S}\right| \sqrt{\frac{\log\left(\frac{1}{\delta}\right) + \log\left(2\left|\mathcal{S}\right|^2 \left|\mathcal{A}\right|\right)}{K d_\mu^{M_\mathcal{D}}}}, \tag{108}$$

where K is the number of trajectories in the dataset $\mathcal{D}$ and $d_\mu^{M_\mathcal{D}}$ is the minimal visitation state-action distribution induced by the behavior policy μ in $M_\mathcal{D}$ (see Eq. (16)).

Proof. $\forall s, s' \in \mathcal{S}, a \in \mathcal{A}$ with $\rho_\pi^{M_\mathcal{D}}(s,a) > 0$, note that $\rho_\pi^{M_\mathcal{D}}(s,a) \ge d_\mu^{M_\mathcal{D}}$ and according to Binomial theorem and Hoeffding's inequality, $\forall \epsilon \in [0,1]$,

$$\begin{aligned}
\mathbb{P}\left(\left|P^{M_\mathcal{D}}(s'|s,a) - P^M(s'|s,a)\right| \ge \epsilon\right) &\le 2\left(1 - d_\mu^{M_\mathcal{D}} + d_\mu^{M_\mathcal{D}} \exp\left(-2\epsilon^2\right)\right)^K && (109)\\
&\le 2\left(1 - d_\mu^{M_\mathcal{D}} \epsilon^2\right)^K && (110)
\end{aligned}$$

and

$$\mathbb{P}\left(\left|r^{M_\mathcal{D}}(s,a) - r^M(s,a)\right| \ge \epsilon\right) \le 2\left(1 - d_\mu^{M_\mathcal{D}} \epsilon^2\right)^K. \tag{111}$$

Thus, using union bound, $\forall \delta \in (0,1]$, with probability $1-\delta$, denote

$$\epsilon_P = \max_{s\in\mathcal{S}, a\in\mathcal{A} \text{ with } \rho_\pi^{M_\mathcal{D}}(s,a)>0} \left\| P^{M_\mathcal{D}}(\cdot|s,a) - P^M(\cdot|s,a) \right\|_1 \tag{112}$$

$$\leq \max_{s\in\mathcal{S}, a\in\mathcal{A} \text{ with } \rho_\pi^{M_\mathcal{D}}(s,a)>0} |\mathcal{S}| \left\| P^{M_\mathcal{D}}(\cdot|s,a) - P^M(\cdot|s,a) \right\|_\infty \tag{113}$$

$$\leq |\mathcal{S}| \sqrt{\frac{\log\left(\frac{1}{\delta}\right) + \log\left(2|\mathcal{S}|^2|\mathcal{A}|\right)}{K d_\mu^{M_\mathcal{D}}}}, \tag{114}$$

$$\epsilon_r = \max_{s\in\mathcal{S}, a\in\mathcal{A} \text{ with } \rho_\pi^{M_\mathcal{D}}(s,a)>0} \left| r^{M_\mathcal{D}}(s,a) - r^M(s,a) \right| \leq \sqrt{\frac{\log\left(\frac{1}{\delta}\right) + \log\left(2|\mathcal{S}||\mathcal{A}|\right)}{K d_\mu^{M_\mathcal{D}}}}, \tag{115}$$

and $r_{max} = 1$, thus, according to Lemma 2 (a variant of the simulation lemma in offline RL), we have

$$|\mathcal{J}_{M_\mathcal{D}}(\pi) - \mathcal{J}_M(\pi)| \leq \left(H + \frac{H(H-1)}{2}|\mathcal{S}|\right) \sqrt{\frac{\log\left(\frac{1}{\delta}\right) + \log\left(2|\mathcal{S}|^2|\mathcal{A}|\right)}{K d_\mu^{M_\mathcal{D}}}} \tag{116}$$

$$\leq H^2 |\mathcal{S}| \sqrt{\frac{\log\left(\frac{1}{\delta}\right) + \log\left(2|\mathcal{S}|^2|\mathcal{A}|\right)}{K d_\mu^{M_\mathcal{D}}}} \tag{117}$$

□

Lemma 1. *In an MDP M, for each behavior policy μ and batch-constrained policy π, collect a dataset $\mathcal{D}$ and the gap between approximate offline policy evaluation $\mathcal{J}_\mathcal{D}(\pi)$ and accurate policy evaluation $\mathcal{J}_M(\pi)$ will asymptotically approach to 0, as the offline dataset $\mathcal{D}$ grows.*

Proof. From Lemma 3, as the size of an offline dataset $|\mathcal{D}| = KH$ grows, the gap between approximate offline policy evaluation $\mathcal{J}_\mathcal{D}(\pi)$ and accurate policy evaluation $\mathcal{J}_M(\pi)$ will asymptotically approach to 0. □

A.7. Omitted Lemmas for Theorem 3

Lemma 4. *In an MDP M and an arbitrary offline dataset $\mathcal{D}$, for each policy π,*

$$p_{out}^{M,\mathcal{D}}(\pi) \leq H\left(\left\|\rho_\pi^M(s) - \rho_\pi^{M_\mathcal{D}}(s)\right\|_1 + \left\|\rho_\pi^M(s,a,r) - \rho_\pi^{M_\mathcal{D}}(s,a,r)\right\|_1\right), \tag{118}$$

where $\rho_\pi^M(s,a,r)$ is the visitation distribution of (s,a,r) in M, as defined in Eq. (9).

Proof. For each τ_H leaving $\mathcal{D}$, we use the first outlier data ($s_t \notin \mathcal{D}$ or $(s_t,a_t,r_t) \notin \mathcal{D}$) to represent τ_H. Thus,

$$p_{out}^{M,\mathcal{D}}(\pi) = \sum_{\tau_H} p_\pi^M(\tau_H)\, \mathbb{1}\left[\exists t \in [H] \text{ s.t. } s_t \notin \mathcal{D} \text{ or } (s_t,a_t,r_t) \notin \mathcal{D}\right] \tag{119}$$

$$\leq H\left(\sum_{s\in\mathcal{S} \text{ with } \rho_\pi^{M_\mathcal{D}}(s)=0} \rho_\pi^M(s) + \sum_{s\in\mathcal{S},a\in\mathcal{A},r\in\mathcal{R} \text{ with } \rho_\pi^{M_\mathcal{D}}(s,a,r)=0} \rho_\pi^M(s,a,r)\right) \tag{120}$$

$$\leq H\left(\left\|\rho_\pi^M(s) - \rho_\pi^{M_\mathcal{D}}(s)\right\|_1 + \left\|\rho_\pi^M(s,a,r) - \rho_\pi^{M_\mathcal{D}}(s,a,r)\right\|_1\right). \tag{121}$$

□

Lemma 5. *In a transformed BAMDP $\overline{M}^+$ with an offline multi-task dataset $\overline{\mathcal{D}}^+$ collected by task-dependent behavior policies $[\mu]$, denoting the i.i.d. offline meta-training tasks sampled from $p(\kappa)$ in $\overline{\mathcal{D}}^+$ by $\mathcal{K}_{train}$, for each meta-testing task $\kappa_{test} \sim p(\kappa)$, and denoting the closest offline meta-training task to κ_{test} by κ_{i^*}, i.e.,*

$$\kappa_{i^*} = \underset{\kappa_i \in \mathcal{K}_{train}}{\arg\min} \left\|\kappa_i - \kappa_{test}\right\|_\infty \tag{122}$$

$$= \underset{\kappa_i \in \mathcal{K}_{train}}{\arg\min} \max\left(\left\|P^{\kappa_i}(s,a,s') - P^{\kappa_{test}}(s,a,s')\right\|_\infty, \left\|R^{\kappa_i}(s,a,r) - R^{\kappa_{test}}(s,a,r)\right\|_\infty\right), \tag{123}$$

then for any batch-constrained policy π in $\mathcal{D}_{\kappa_{i^*},\mu_{i^*}}$, where $\mathcal{D}_{\kappa_{i^*},\mu_{i^*}}$ is a sub-dataset collected in $\overline{\mathcal{D}}^+$ (see Definition 3), i.e., $\forall \pi \in \Pi^{\mathcal{D}_{\kappa_{i^*},\mu_{i^*}}}$,

$$p_{out}^{\kappa_{test},\mathcal{D}_{\kappa_{i^*},\mu_{i^*}}}(\pi) \tag{124}$$

$$\leq 2H^2 |\mathcal{S}|^2 |\mathcal{A}| |\mathcal{R}| \left(\underbrace{\sqrt{\frac{\log\left(\frac{1}{\delta}\right) + \log\left(2|\mathcal{S}|^2|\mathcal{A}|\right)}{K_{\kappa_{i^*},\mu_{i^*}} \cdot d_\mu^{M_{\mathcal{D}_{\kappa_{i^*},\mu_{i^*}}}}}}}_{\substack{\textit{Asymptotically approaches zero,}\\ \textit{when } \mathcal{D}_{\kappa_{i^*},\mu_{i^*}} \textit{ is sufficiently large}}} + \underbrace{2\left(\frac{\log\left(\frac{1}{\delta}\right)}{|\mathcal{K}_{train}|}\right)^{\frac{1}{|\mathcal{S}||\mathcal{A}|(|\mathcal{S}|+|\mathcal{R}|)}}}_{\substack{\textit{Asymptotically approaches zero,}\\ \textit{when } \mathcal{K}_{train} \textit{ is sufficiently large}}}\right), \tag{125}$$

where $K_{\kappa_{i^*},\mu_{i^*}}$ is the number of trajectories in the sub-dataset $\mathcal{D}_{\kappa_{i^*},\mu_{i^*}}$, $d_\mu^{M_{\mathcal{D}_{\kappa_{i^*},\mu_{i^*}}}}$ is the minimal visitation state-action distribution induced by the behavior policy μ in $M_{\mathcal{D}_{\kappa_{i^*},\mu_{i^*}}}$ (see Eq. (16)), and $|\mathcal{K}_{train}|$ is the number of i.i.d. offline meta-training tasks sampled from $p(\kappa)$ in $\overline{\mathcal{D}}^+$.

Proof. According to Lemma 4,

$$p_{out}^{\kappa_{test},\mathcal{D}_{\kappa_{i^*},\mu_{i^*}}}(\pi) \tag{126}$$

$$\leq H\left(\left\|\rho_\pi^{M_{\mathcal{D}_{\kappa_{i^*},\mu_{i^*}}}}(s) - \rho_\pi^{\kappa_{test}}(s)\right\|_1 + \left\|\rho_\pi^{M_{\mathcal{D}_{\kappa_{i^*},\mu_{i^*}}}}(s,a,r) - \rho_\pi^{\kappa_{test}}(s,a,r)\right\|_1\right) \tag{127}$$

$$\leq H\left(\left\|\rho_\pi^{M_{\mathcal{D}_{\kappa_{i^*},\mu_{i^*}}}}(s) - \rho_\pi^{\kappa_{i^*}}(s)\right\|_1 + \left\|\rho_\pi^{M_{\mathcal{D}_{\kappa_{i^*},\mu_{i^*}}}}(s,a,r) - \rho_\pi^{\kappa_{i^*}}(s,a,r)\right\|_1\right) + \tag{128}$$

$$H\left(\left\|\rho_\pi^{\kappa_{i^*}}(s) - \rho_\pi^{\kappa_{test}}(s)\right\|_1 + \left\|\rho_\pi^{\kappa_{i^*}}(s,a,r) - \rho_\pi^{\kappa_{test}}(s,a,r)\right\|_1\right). \tag{129}$$

Part I Similar with Lemma 3, $\forall\delta\in(0,1]$, with probability $1-\delta$, denote

$$\epsilon_P = \max_{s\in\mathcal{S},a\in\mathcal{A} \text{ with } \rho_\pi^{M_{\mathcal{D}_{\kappa_{i^*}},\mu_{i^*}}}(s,a)>0} \left\| P^{M_{\mathcal{D}_{\kappa_{i^*}},\mu_{i^*}}}(\cdot|s,a) - P^{\kappa_{i^*}}(\cdot|s,a) \right\|_1 \tag{130}$$

$$\leq |\mathcal{S}| \sqrt{\frac{\log\left(\frac{1}{\delta}\right) + \log\left(2|\mathcal{S}|^2|\mathcal{A}|\right)}{K_{\kappa_{i^*},\mu_{i^*}} \cdot d_\mu^{M_{\mathcal{D}_{\kappa_{i^*}},\mu_{i^*}}}}}, \tag{131}$$

$$\epsilon_r = \max_{s\in\mathcal{S},a\in\mathcal{A} \text{ with } \rho_\pi^{M_{\mathcal{D}_{\kappa_{i^*}},\mu_{i^*}}}(s,a)>0} \left| r^{M_{\mathcal{D}_{\kappa_{i^*}},\mu_{i^*}}}(s,a) - r^{\kappa_{i^*}}(s,a) \right| \tag{132}$$

$$\leq \sqrt{\frac{\log\left(\frac{1}{\delta}\right) + \log\left(2|\mathcal{S}||\mathcal{A}|\right)}{K_{\kappa_{i^*},\mu_{i^*}} \cdot d_\mu^{M_{\mathcal{D}_{\kappa_{i^*}},\mu_{i^*}}}}}, \tag{133}$$

thus, from Lemma 6, we have

$$H\left(\left\|\rho_\pi^{M_{\mathcal{D}_{\kappa_{i^*}},\mu_{i^*}}}(s) - \rho_\pi^{\kappa_{i^*}}(s)\right\|_1 + \left\|\rho_\pi^{M_{\mathcal{D}_{\kappa_{i^*}},\mu_{i^*}}}(s,a,r) - \rho_\pi^{\kappa_{i^*}}(s,a,r)\right\|_1\right) \tag{134}$$

$$\leq 2H^2|\mathcal{S}|^2|\mathcal{A}||\mathcal{R}| \sqrt{\frac{\log\left(\frac{1}{\delta}\right) + \log\left(2|\mathcal{S}|^2|\mathcal{A}|\right)}{K_{\kappa_{i^*},\mu_{i^*}} \cdot d_\mu^{M_{\mathcal{D}_{\kappa_{i^*}},\mu_{i^*}}}}}. \tag{135}$$

Part II From Lemma 10, $\forall\delta\in(0,1]$, with probability $1-\delta$, denote

$$\tilde{\epsilon}_P = \max_{s\in\mathcal{S},a\in\mathcal{A}} \|P^{\kappa_{i^*}}(\cdot|s,a) - P^{\kappa_{test}}(\cdot|s,a)\|_1 \tag{136}$$

$$\leq |\mathcal{S}| \|\kappa_{i^*} - \kappa_{test}\|_\infty \tag{137}$$

$$\leq 2|\mathcal{S}| \left(\frac{\log\left(\frac{1}{\delta}\right)}{|\mathcal{K}_{train}|}\right)^{\frac{1}{|\mathcal{S}||\mathcal{A}|(|\mathcal{S}|+|\mathcal{R}|)}}, \tag{138}$$

$$\tilde{\epsilon}_R = \max_{s\in\mathcal{S},a\in\mathcal{A},r\in\mathcal{R}} |R^{\kappa_{i^*}}(r|s,a) - R^{\kappa_{test}}(r|s,a)| \tag{139}$$

$$\leq \|\kappa_{i^*} - \kappa_{test}\|_\infty \tag{140}$$

$$\leq 2\left(\frac{\log\left(\frac{1}{\delta}\right)}{|\mathcal{K}_{train}|}\right)^{\frac{1}{|\mathcal{S}||\mathcal{A}|(|\mathcal{S}|+|\mathcal{R}|)}}, \tag{141}$$

then from Lemma 8, we have

$$H\left(\|\rho_\pi^{\kappa_{i^*}}(s) - \rho_\pi^{\kappa_{test}}(s)\|_1 + \|\rho_\pi^{\kappa_{i^*}}(s,a,r) - \rho_\pi^{\kappa_{test}}(s,a,r)\|_1\right) \tag{142}$$

$$\leq 4H^2|\mathcal{S}|^2|\mathcal{A}||\mathcal{R}| \left(\frac{\log\left(\frac{1}{\delta}\right)}{|\mathcal{K}_{train}|}\right)^{\frac{1}{|\mathcal{S}||\mathcal{A}|(|\mathcal{S}|+|\mathcal{R}|)}}. \tag{143}$$

Overall Combining Part I and Part II, we have

$$p_{out}^{\kappa_{test},\mathcal{D}_{\kappa_{i^*}},\mu_{i^*}}(\pi) \tag{144}$$

$$\leq H\left(\left\|\rho_\pi^{M_{\mathcal{D}_{\kappa_{i^*}},\mu_{i^*}}}(s) - \rho_\pi^{\kappa_{i^*}}(s)\right\|_1 + \left\|\rho_\pi^{M_{\mathcal{D}_{\kappa_{i^*}},\mu_{i^*}}}(s,a,r) - \rho_\pi^{\kappa_{i^*}}(s,a,r)\right\|_1\right) + \tag{145}$$

$$H\left(\|\rho_\pi^{\kappa_{i^*}}(s) - \rho_\pi^{\kappa_{test}}(s)\|_1 + \|\rho_\pi^{\kappa_{i^*}}(s,a,r) - \rho_\pi^{\kappa_{test}}(s,a,r)\|_1\right) \tag{146}$$

$$\leq 2H^2|\mathcal{S}|^2|\mathcal{A}||\mathcal{R}| \left(\sqrt{\frac{\log\left(\frac{1}{\delta}\right) + \log\left(2|\mathcal{S}|^2|\mathcal{A}|\right)}{K_{\kappa_{i^*},\mu_{i^*}} \cdot d_\mu^{M_{\mathcal{D}_{\kappa_{i^*}},\mu_{i^*}}}}} + 2\left(\frac{\log\left(\frac{1}{\delta}\right)}{|\mathcal{K}_{train}|}\right)^{\frac{1}{|\mathcal{S}||\mathcal{A}|(|\mathcal{S}|+|\mathcal{R}|)}}\right). \tag{147}$$

□

A.7.1. DETAILED LEMMAS (PART I)

Lemma 6. *In an MDP M with an offline dataset $\mathcal{D}$, for any batch-constrained policy $\pi \in \Pi^{\mathcal{D}}$, if*

$$\max_{s\in\mathcal{S},a\in\mathcal{A} \text{ with } \rho_\pi^{M_\mathcal{D}}(s,a)>0} \left\|P^{M_\mathcal{D}}(\cdot|s,a) - P^M(\cdot|s,a)\right\|_1 \le \epsilon_P, \tag{148}$$

$$\max_{s\in\mathcal{S},a\in\mathcal{A},r\in\mathcal{R} \text{ with } \rho_\pi^{M_\mathcal{D}}(s,a)>0} \left|R^{M_\mathcal{D}}(r|s,a) - R^M(r|s,a)\right| \le \epsilon_R, \tag{149}$$

we have

$$\left\|\rho_\pi^M(s) - \rho_\pi^{M_\mathcal{D}}(s)\right\|_1 + \left\|\rho_\pi^M(s,a,r) - \rho_\pi^{M_\mathcal{D}}(s,a,r)\right\|_1 \tag{150}$$

$$\le |\mathcal{S}|\left(|\mathcal{A}|\,|\mathcal{R}|+1\right)\frac{H-1}{2}\epsilon_P + |\mathcal{S}|\,|\mathcal{A}|\,|\mathcal{R}|\,\epsilon_R. \tag{151}$$

Proof. For each $\hat{s}\in\mathcal{S}, \hat{a}\in\mathcal{A}$, create an auxiliary reward function $\tilde{r}(s,a):\mathcal{S}\times\mathcal{A}\to[0,1]$: $\forall s\in\mathcal{S}, a\in\mathcal{A}$,

$$\tilde{r}(s,a) = \begin{cases} \frac{1}{H}, & \text{if } s=\hat{s} \text{ and } a=\hat{a}, \\ 0, & \text{otherwise.} \end{cases} \tag{152}$$

Denote $\widehat{M} = \left(\mathcal{S},\mathcal{A},\mathcal{R},H,P^M,\tilde{r}\right)$ and $\widehat{M}_\mathcal{D} = \left(\mathcal{S},\mathcal{A},\mathcal{R},H,P^{M_\mathcal{D}},\tilde{r}\right)$. Using the offline Simulation Lemma shown in Lemma 2, for any batch-constrained policy $\pi\in\Pi^{\mathcal{D}}$,

$$\rho_\pi^M(\hat{s},\hat{a}) = \mathcal{J}_{\widehat{M}}(\pi) \quad \text{and} \quad \rho_\pi^{M_\mathcal{D}}(\hat{s},\hat{a}) = \mathcal{J}_{\widehat{M}_\mathcal{D}}(\pi). \tag{153}$$

Thus, $\epsilon_r = 0, r_{max} = \frac{1}{H}$, and

$$\left|\rho_\pi^M(\hat{s},\hat{a}) - \rho_\pi^{M_\mathcal{D}}(\hat{s},\hat{a})\right| \le \left|\mathcal{J}_{\widehat{M}}(\pi) - \mathcal{J}_{\widehat{M}_\mathcal{D}}(\pi)\right| \tag{154}$$

$$\le H\epsilon_r + \frac{H(H-1)r_{max}}{2}\epsilon_P \tag{155}$$

$$= \frac{H-1}{2}\epsilon_P. \tag{156}$$

Similarly, we have $\forall s\in\mathcal{S}$,

$$\left|\rho_\pi^M(s) - \rho_\pi^{M_\mathcal{D}}(s)\right| \le \frac{H-1}{2}\epsilon_P. \tag{157}$$

For any $s\in\mathcal{S}, a\in\mathcal{A}, r\in\mathcal{R}$,

$$\left|\rho_\pi^M(s,a,r) - \rho_\pi^{M_\mathcal{D}}(s,a,r)\right| \tag{158}$$

$$= \left|\rho_\pi^M(s,a)R^M(r|s,a) - \rho_\pi^{M_\mathcal{D}}(s,a)R^{M_\mathcal{D}}(r|s,a)\right| \tag{159}$$

$$\le \left|\rho_\pi^M(s,a) - \rho_\pi^{M_\mathcal{D}}(s,a)\right| R^M(r|s,a) + \rho_\pi^{M_\mathcal{D}}(s,a)\left|R^M(r|s,a) - R^{M_\mathcal{D}}(r|s,a)\right| \tag{160}$$

$$\le \left|\rho_\pi^M(s,a) - \rho_\pi^{M_\mathcal{D}}(s,a)\right| + \left|R^M(r|s,a) - R^{M_\mathcal{D}}(r|s,a)\right| \tag{161}$$

$$\le \frac{H-1}{2}\epsilon_P + \epsilon_R. \tag{162}$$

Therefore,

$$\left\|\rho_\pi^M(s) - \rho_\pi^{M_\mathcal{D}}(s)\right\|_1 + \left\|\rho_\pi^M(s,a,r) - \rho_\pi^{M_\mathcal{D}}(s,a,r)\right\|_1 \tag{163}$$

$$\le |\mathcal{S}|\frac{H-1}{2}\epsilon_P + |\mathcal{S}|\,|\mathcal{A}|\,|\mathcal{R}|\left(\frac{H-1}{2}\epsilon_P + \epsilon_R\right) \tag{164}$$

$$= |\mathcal{S}|\left(|\mathcal{A}|\,|\mathcal{R}|+1\right)\frac{H-1}{2}\epsilon_P + |\mathcal{S}|\,|\mathcal{A}|\,|\mathcal{R}|\,\epsilon_R. \tag{165}$$

□

A.7.2. DETAILED LEMMAS (PART II)

Lemma 7 (Simulation Lemma for Finite-Horizon MDPs)**.** *Given a pair of finite-horizon MDPs M_1 and M_2 with the same state space $\mathcal{S}$, same action space $\mathcal{A}$, same reward function $\mathcal{R}$, and same horizon H. If*

$$\max_{s\in\mathcal{S},a\in\mathcal{A}} \left\|P^{M_1}(\cdot|s,a) - P^{M_2}(\cdot|s,a)\right\|_1 \leq \tilde{\epsilon}_P, \tag{166}$$

$$\max_{s\in\mathcal{S},a\in\mathcal{A}} \left|r^{M_1}(s,a) - r^{M_2}(s,a)\right| \leq \tilde{\epsilon}_r, \tag{167}$$

$$\max_{s\in\mathcal{S},a\in\mathcal{A}} \max\left(r^{M_1}(s,a), r^{M_2}(s,a)\right) \leq r_{max}, \tag{168}$$

where $r^{M_1}(s,a) = \mathbb{E}_{\tilde{r}\sim R^{M_1}(s,a)}[\tilde{r}]$ and $r^{M_2}(s,a) = \mathbb{E}_{\tilde{r}\sim R^{M_2}(s,a)}[\tilde{r}]$, we have

$$|\mathcal{J}_{M_1}(\pi) - \mathcal{J}_{M_2}(\pi)| \leq H\tilde{\epsilon}_r + \frac{H(H-1)r_{max}}{2}\tilde{\epsilon}_P. \tag{169}$$

Proof. Similar with famous Simulation Lemma in finite-horizon MDPs (Alekh Agarwal, 2017) and the offline variant shown in Lemma 2, we will prove $\forall h \in [H], \forall s \in \mathcal{S}_h$,

$$\left|V_\pi^{M_1}(s) - V_\pi^{M_2}(s)\right| \leq (H-h)\tilde{\epsilon}_r + \frac{(H-h)(H-h-1)r_{max}}{2}\tilde{\epsilon}_P \tag{170}$$

by induction. When $h = H-1$, we have $\forall s \in \mathcal{S}_h$, $\left|V_\pi^{M_1}(s) - V_\pi^{M_2}(s)\right| \leq \tilde{\epsilon}_r$ holds. And $\forall h \in [H-1]$, $\forall s \in \mathcal{S}_h$,

$$\left|V_\pi^{M_1}(s) - V_\pi^{M_2}(s)\right| \tag{171}$$

$$\leq \sum_{a\in\mathcal{A}} \pi(a|s)\left|r^{M_1}(s,a) - r^{M_2}(s,a)\right| + \tag{172}$$

$$\sum_{a\in\mathcal{A}} \pi(a|s) \sum_{s'\in\mathcal{S}_{h+1}} \left|P^{M_1}(s'|s,a)V_\pi^{M_1}(s') - P^{M_2}(s'|s,a)V_\pi^{M_2}(s')\right| \tag{173}$$

$$\leq (H-h)\tilde{\epsilon}_r + \frac{(H-h)(H-h-1)r_{max}}{2}\tilde{\epsilon}_P. \tag{174}$$

Thus,

$$|\mathcal{J}_{M_1}(\pi) - \mathcal{J}_{M_2}(\pi)| = \left|V_\pi^{M_1}(s_0) - V_\pi^{M_2}(s_0)\right| \tag{175}$$

$$\leq H\tilde{\epsilon}_r + \frac{H(H-1)r_{max}}{2}\tilde{\epsilon}_P. \tag{176}$$

□

Lemma 8. *Given a pair of finite-horizon MDPs M_1 and M_2 with the same state space, same action space, same reward function, and same horizon. If*

$$\max_{s\in\mathcal{S},a\in\mathcal{A}} \left\|P^{M_1}(\cdot|s,a) - P^{M_2}(\cdot|s,a)\right\|_1 \leq \tilde{\epsilon}_P, \tag{177}$$

$$\max_{s\in\mathcal{S},a\in\mathcal{A},r\in\mathcal{R}} \left|R^{M_1}(r|s,a) - R^{M_2}(r|s,a)\right| \leq \tilde{\epsilon}_R, \tag{178}$$

we have

$$\left\|\rho_\pi^{M_1}(s) - \rho_\pi^{M_2}(s)\right\|_1 + \left\|\rho_\pi^{M_1}(s,a,r) - \rho_\pi^{M_2}(s,a,r)\right\|_1 \tag{179}$$

$$\leq |\mathcal{S}|\left(|\mathcal{A}|\,|\mathcal{R}| + 1\right)\frac{H-1}{2}\tilde{\epsilon}_P + |\mathcal{S}|\,|\mathcal{A}|\,|\mathcal{R}|\,\tilde{\epsilon}_R. \tag{180}$$

Proof. Similar with Lemma 6, $\forall s \in \mathcal{S}, a \in \mathcal{A}, r \in \mathcal{R}$, using Simulation Lemma 7, we have

$$\left|\rho_\pi^{M_1}(s) - \rho_\pi^{M_2}(s)\right| \leq \frac{H-1}{2}\tilde{\epsilon}_P \quad \text{and} \quad \left|\rho_\pi^{M_1}(s,a) - \rho_\pi^{M_2}(s,a)\right| \leq \frac{H-1}{2}\tilde{\epsilon}_P, \tag{181}$$

and

$$\left|\rho_\pi^M(s,a,r) - \rho_\pi^{M_\mathcal{D}}(s,a,r)\right| \leq \frac{H-1}{2}\tilde{\epsilon_P} + \tilde{\epsilon}_R. \tag{182}$$

Therefore,

$$\left\|\rho_\pi^{M_1}(s) - \rho_\pi^{M_2}(s)\right\|_1 + \left\|\rho_\pi^{M_1}(s,a,r) - \rho_\pi^{M_2}(s,a,r)\right\|_1 \tag{183}$$

$$\leq |\mathcal{S}|\left(|\mathcal{A}|\,|\mathcal{R}|+1\right)\frac{H-1}{2}\tilde{\epsilon}_P + |\mathcal{S}|\,|\mathcal{A}|\,|\mathcal{R}|\,\tilde{\epsilon}_R. \tag{184}$$

□

Lemma 9. *Let X, Y be two i.i.d. random vectors that take values in $[0,1]^n$, $n \in \mathbb{N}^+$. For any $\epsilon \in (0,1]$, we have*

$$\mathbb{P}\left[\max_{i\in[n]} |X_i - Y_i| \geq \epsilon\right] \leq 1 - \left(\frac{\epsilon}{2}\right)^n. \tag{185}$$

Proof. Denote an auxiliary set

$$V = \left\{x \in \mathbb{R}^n \middle| \max_{i\in[n]} |x_i| < \frac{\epsilon}{2}\right\}, \tag{186}$$

then if $X, Y \in V$, we must have

$$\max_{i\in[n]} |X_i - Y_i| < \epsilon. \tag{187}$$

For any $c \in \mathbb{N}^n$, denote

$$V^c = V + v^c, \quad \text{where} \quad v_i^c = \left(c_i + \frac{1}{2}\right)\epsilon, \quad \forall i \in [n]. \tag{188}$$

We may construct a set of such cosets of V as follows:

$$S = \{V^c | c \in C\}, \quad \text{where} \quad C = \left\{c \in \mathbb{N}^n \middle| c_i \in \left[\left\lceil \frac{1}{\epsilon} \right\rceil\right]\right\}. \tag{189}$$

There are several properties related to these constructions:

- For any $c \in \mathbb{N}^n$, if $X, Y \in V^c$, $\max_{i\in[n]} |X_i - Y_i| < \epsilon$.
- The union of sets in S contains $[0,1]^n$
- Any two different sets in S are disjoint.

The only loophole is that we have not considered points in boundaries ∂V^c ($V^c \in S$). These boundaries can be decomposed into disjoint union of hyperplanes in $\mathbb{R}^n$. For each one of the hyperplanes, arbitrarily designate it to an adjacent $V^c \in S$. New V^cs are the union of the original one and the hyperplanes designated to it. Note that

$$\sum_{c\in C} [X \in V^c] = 1. \tag{190}$$

Therefore,

$$\mathbb{P}\left[\max_{i\in[n]} |X_i - Y_i| \geq \epsilon\right] \leq 1 - \sum_{c\in C} \mathbb{P}\left[X \in V^c\right]\mathbb{P}\left[Y \in V^c\right] \tag{191}$$

$$= 1 - \sum_{c\in C} \mathbb{P}\left[X \in V^c\right]^2 \tag{192}$$

$$\leq 1 - \frac{1}{|C|}\left(\sum_{c\in C}\mathbb{P}\left[X \in V^c\right]\right)^2 \tag{193}$$

$$= 1 - \frac{1}{|C|}. \tag{194}$$

Since $\frac{1}{\epsilon} \geq 1$, we have

$$|C| = \left\lceil \frac{1}{\epsilon} \right\rceil^n < \left(1 + \frac{1}{\epsilon}\right)^n \leq \left(\frac{2}{\epsilon}\right)^n \quad \text{and} \quad \mathbb{P}\left[\max_{i\in[n]} |X_i - Y_i| \geq \epsilon\right] \leq 1 - \left(\frac{\epsilon}{2}\right)^n. \tag{195}$$

□

Lemma 10. *In a transformed BAMDP $\overline{M}^+$ with an offline multi-task dataset $\overline{\mathcal{D}}^+$, for any meta-testing task $\kappa_{test} \sim p(\kappa)$, $\forall \delta \in (0, 1]$, with probability $1 - \delta$, we have*

$$\|\kappa_{i^*} - \kappa_{test}\|_\infty \tag{196}$$

$$= \max\left(\|P^{\kappa_{i^*}}(s, a, s') - P^{\kappa_{test}}(s, a, s')\|_\infty, \|R^{\kappa_{i^*}}(s, a, r) - R^{\kappa_{test}}(s, a, r)\|_\infty\right) \tag{197}$$

$$\leq 2\left(\frac{\log\left(\frac{1}{\delta}\right)}{|\mathcal{K}_{train}|}\right)^{\frac{1}{|\mathcal{S}||\mathcal{A}|(|\mathcal{S}|+|\mathcal{R}|)}}, \tag{198}$$

where $\kappa_{i^*} \in \mathcal{K}_{train}$ is the closest offline meta-training task to κ_{test} (see Eq. (122)), $\|\kappa_{i^*} - \kappa_{test}\|_\infty$ is the distance between κ_{i^*} and κ_{test} (see Eq. (123)), and $\mathcal{K}_{train}$ is the i.i.d. offline meta-training tasks sampled from $p(\kappa)$ in $\overline{\mathcal{D}}^+$.

Proof. From Lemma 9, we set $n = |\mathcal{S}|\,|\mathcal{A}|\,(|\mathcal{S}| + |\mathcal{R}|)$, then $\forall \epsilon \in (0, 1], \forall \kappa_i \in \mathcal{K}_{train}$,

$$\mathbb{P}\left[\|\kappa_i - \kappa_{test}\|_\infty \geq \epsilon\right] \leq 1 - \left(\frac{\epsilon}{2}\right)^n. \tag{199}$$

Hence

$$\mathbb{P}\left[\operatorname*{arg\,min}_{\kappa_i\in\mathcal{K}_{train}} \|\kappa_i - \kappa_{test}\|_\infty \geq \epsilon\right] = \prod_{\kappa_i\in\mathcal{K}_{train}} \mathbb{P}\left[\|\kappa_i - \kappa_{test}\|_\infty \geq \epsilon\right] \tag{200}$$

$$\leq \left(1 - \left(\frac{\epsilon}{2}\right)^n\right)^{|\mathcal{K}_{train}|}. \tag{201}$$

Therefore, $\forall \delta \in (0, 1]$, with probability $1 - \delta$,

$$\operatorname*{arg\,min}_{\kappa_i\in\mathcal{K}_{train}} \|\kappa_i - \kappa_{test}\|_\infty \leq 2\left(\frac{\log\left(\frac{1}{\delta}\right)}{|\mathcal{K}_{train}|}\right)^{\frac{1}{|\mathcal{S}||\mathcal{A}|(|\mathcal{S}|+|\mathcal{R}|)}}. \tag{202}$$

□

B. Formal loss function of an Ensemble of Reward and Dynamics Models

In Section 4.2, we will optimize an ensemble of L reward and dynamics models $\{r_{\phi_i}(s,a,z), p_{\psi_i}(s,a,z)\}_{i=1}^{L}$ on the multi-task dataset $\{\mathcal{D}_z\}$ with an latent task embedding z. We will minimize the MSE loss function

$$L_M(\phi,\psi,z) = \mathop{\mathbb{E}}_{(s,a,r,s')\sim\mathcal{D}_z}\left[\frac{1}{L}\sum_{i=1}^{L}\left(\left(r - r^i_\phi(s,a,z)\right)^2 + \left\|s' - p^i_\psi(s,a,z)\right\|_2^2\right)\right] \tag{203}$$

to train the parameters $\{\phi_i,\psi_i\}_{i=1}^{L}$ with the given latent task embedding z.

C. Hyper-Parameter Settings

Environment Settings. Table 4 shows hyper-parameter settings for the task sets used in our experiments. Most hyper-parameters are adopted from previous works (Li et al., 2020b; Mitchell et al., 2021). For all task sets, 80% of the tasks are meta-training tasks, and the remaining 20% tasks are meta-testing tasks.

Table 4. Environment parameter settings.

Environment	Episode Length	# of Adaptation Episodes	# of Tasks	# of Trajectories per Task
All Meta-World Envs	500	10	50	45
Cheetah-Vel	200	10	100	45
Point-Robot	20	20	100	45
Point-Robot-Sparse	20	20	100	45

IDAQ hyper-parameter settings. Table 5 shows IDAQ's hyper-parameter settings. Most hyper-parameters are adopted from FOCAL (Li et al., 2020b). We set n_e to 1 as the evaluation environments are all nearly deterministic.

Table 5. Detailed hyper-parameter settings for IDAQ.

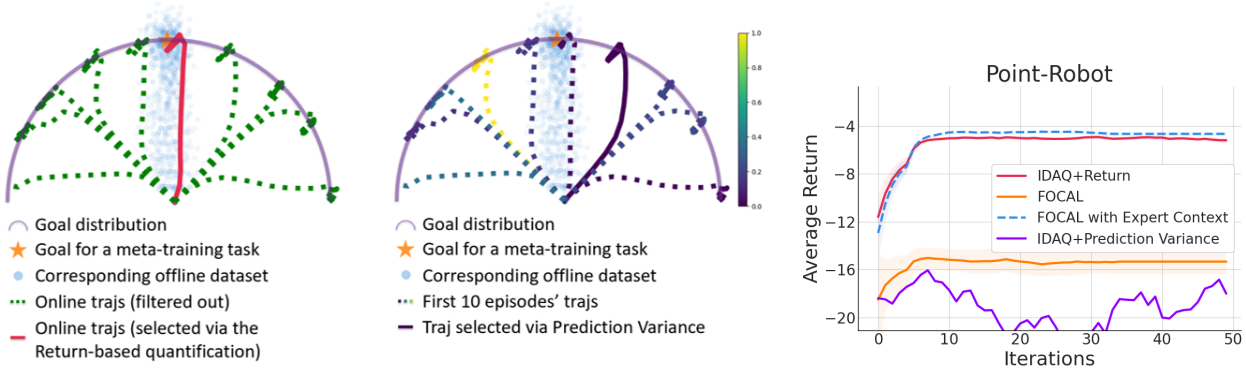
Hyper-Parameter	Hyper-Parameter Values
batch size	256
meta batch size	16
learning rates for dual critic	1e-4
learning rates for all other components	3e-4
network structure for all components	three fully connected layers with 200 units
optimizer	adam
discount	0.99
latent size	20
reward scale	100 for point envs, 1 for all other envs
n_r	1/2 of total adaptation episodes
n_i	1/2 of total adaptation episodes
k	10 for point envs, 20 for all other envs
L	4
n_e	1

D. Didactic Example on Distribution Shift

To empirically demonstrate the distributional shift problem proposed in Section 3, we introduce Point-Robot, a simple 2D navigation task set commonly used in meta-RL (Rakelly et al., 2019; Zhang et al., 2021). Figure 5(a) illustrates the distribution mismatch between offline meta-training and online adaptation, as the dataset is collected by task-dependent behavior policies. As a consequence, directly performing adaptation with the online collected trajectories leads to poor adaptation performance (see FOCAL in Figure 5(c)). Figure 5(b) shows that the **Prediction Variance** quantification cannot correctly detect OOD data, as there is a large error in uncertainty estimation. The **Return-based** quantification fixes this problem by greedily selecting trajectories. As shown in Figure 5(a), at the end of the reference stage, the **Return-based** quantification selects the red trajectory as it has the highest return. After the reference stage, IDAQ iteratively optimizes the posterior belief to get the final policy. As shown in Figure 5(c), IDAQ+Return achieves comparable performance to FOCAL with expert context and significantly outperforms FOCAL with online adaptation.

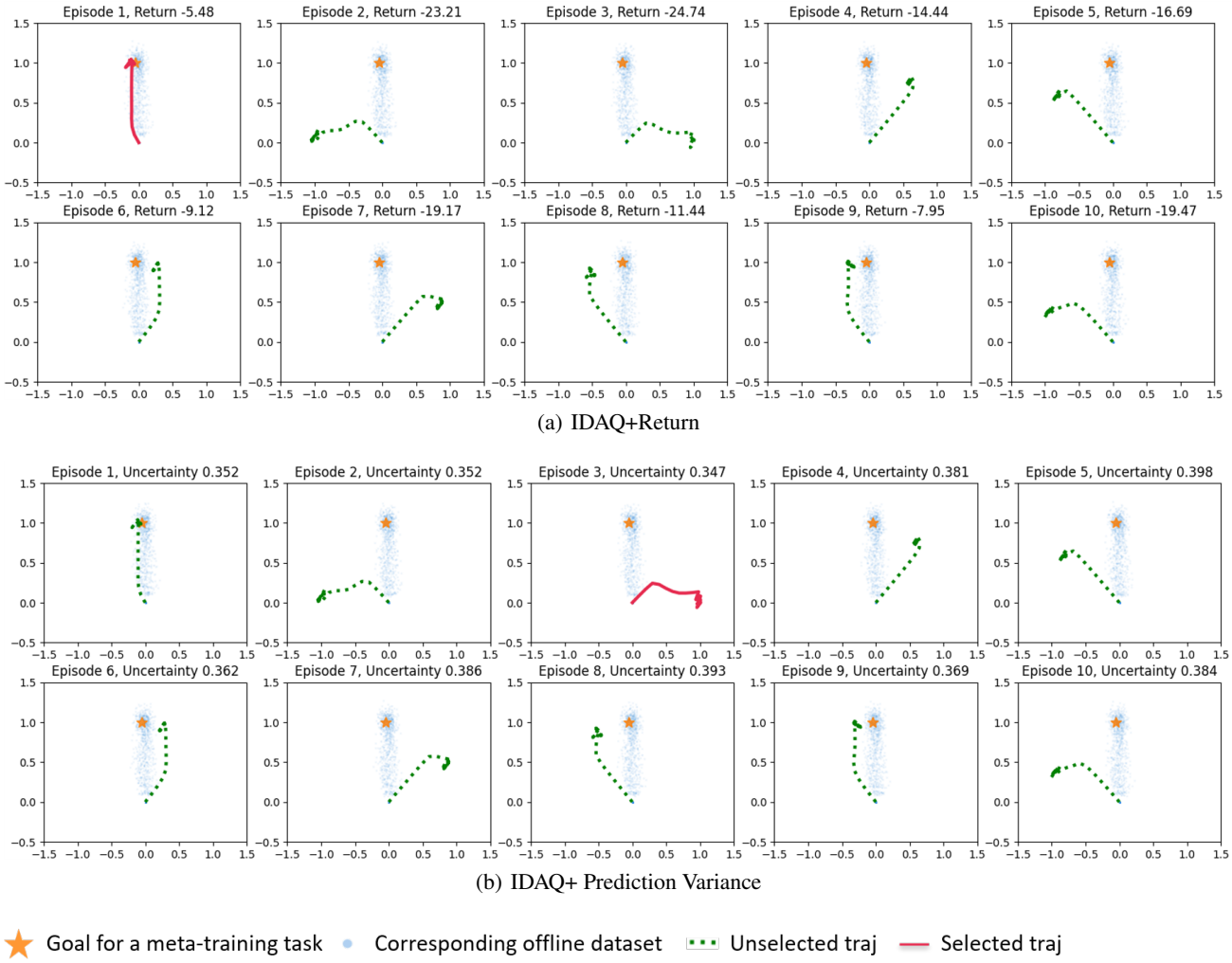
To further investigate why the **Prediction Variance** quantification fails to identify in-distribution trajectories, we demonstrate the ensemble's uncertainty estimation on the first 10 trajectories collected in the reference stage. As shown in Figure 6, the **Prediction Variance** quantification cannot accurately estimate the distance to the offline dataset, and fail to identify in-distribution data. On the other hand, the **Return-based** quantification can successfully select in-distribution data with its greedy selection mechanism.

Figure 7 shows IDAQ+Return, FOCAL, and IDAQ+Prediction Variance's adaptation visualization (Episode 11-20) after the reference phase in adaptation (Episode 1-10). Results demonstrate that while the **Return-based** quantification is able to identify in-distribution data and achieve superior adaptation performance, FOCAL utilizes all the 10 trajectories for adaptation, and cannot correctly update task belief due to the distributional shift problem. The **Prediction Variance** quantification, as discussed above, fails to correctly select the in-distribution trajectory, and thus cannot successfully reach the goal.



(a) IDAQ+Return (b) IDAQ+Prediction Variance (c) Learning Curve

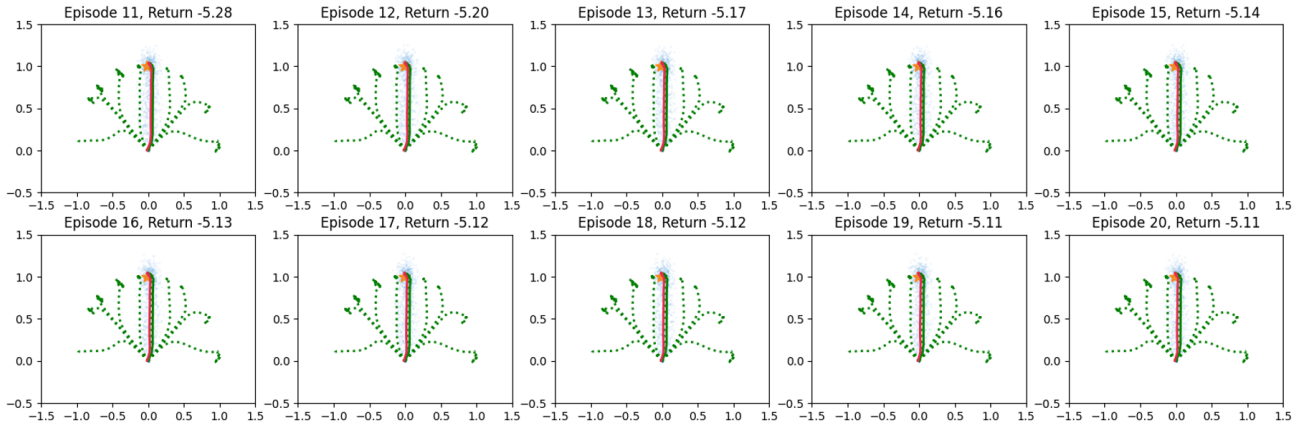
Figure 5. (a) Illustration of data distribution mismatch between offline meta-training (blue) and IDAQ+**Return**'s online adaptation (green and red trajectories). (b) The agent fails to identify in-distribution trajectories via the **Prediction Variance** quantification. Trajectories are colored with corresponding normalized uncertainty estimation. (c) Adaptation performance of IDAQ+Return, FOCAL, FOCAL with expert context, and IDAQ+Prediction Variance.



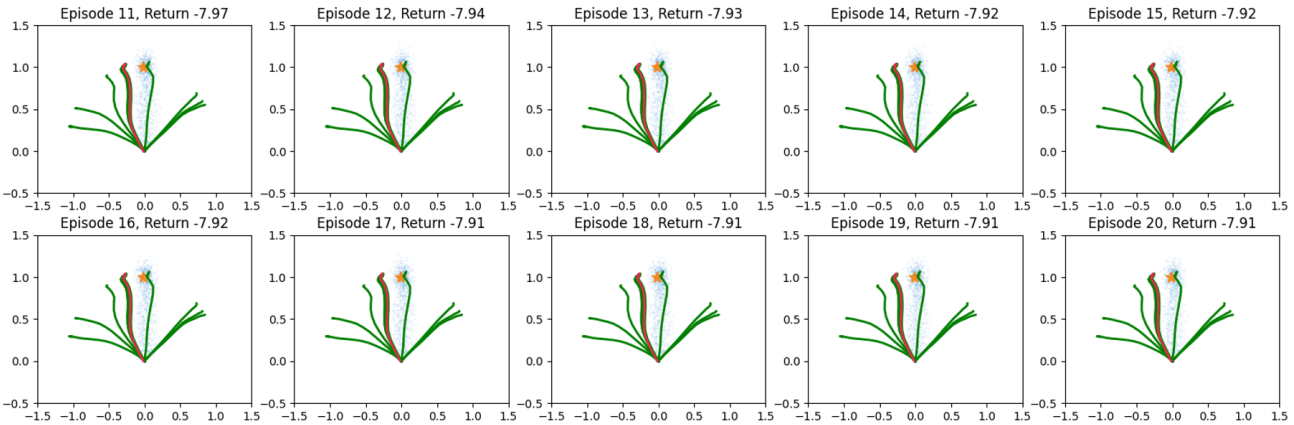
(a) IDAQ+Return

(b) IDAQ+ Prediction Variance

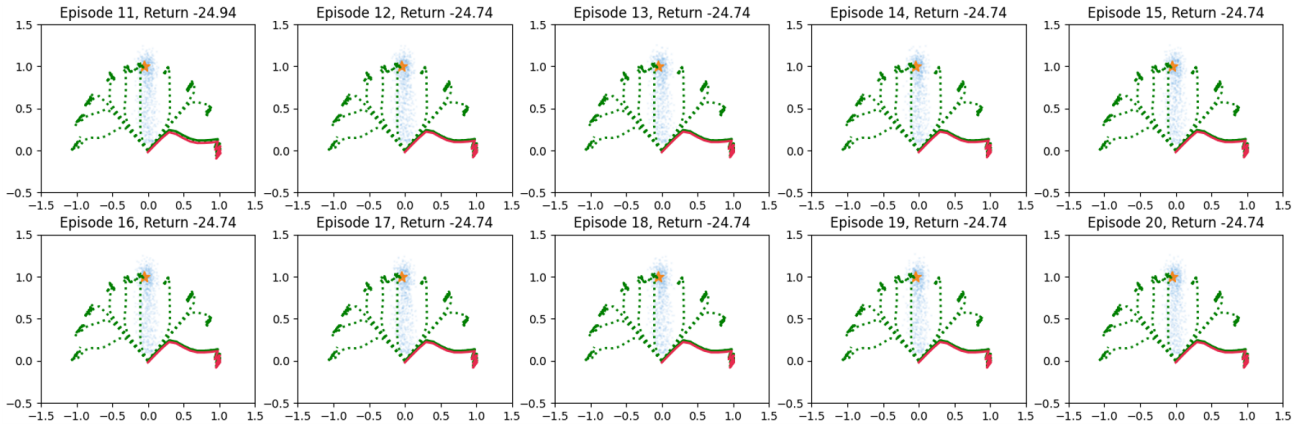
Figure 6. Visualization of the **Return-based** quantification and the **Prediction Variance** quantification's trajectory selection. (a) The **Return-based** quantification successfully selects the in-distribution trajectory. (b) The **Prediction Variance** quantification cannot identify in-distribution data, as its uncertainty estimation is not accurate.



(a) IDAQ+Return



(b) FOCAL



(c) IDAQ+Prediction Variance

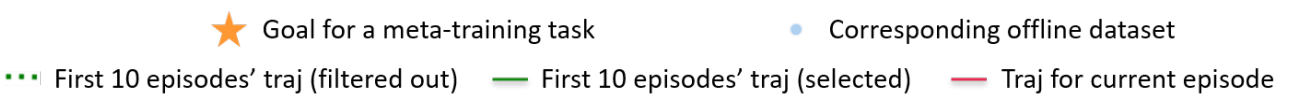


Figure 7. Visualization of IDAQ+Return, FOCAL and IDAQ+Prediction Variance's adaptation in Episode 11-20 on the Point-Robot environment. (a) The **Return-based** quantification successfully selects the in-distribution trajectory and keeps improving in adaptation. (b) FOCAL suffers from the out-of-distribution problem, and cannot correctly update posterior belief, leading to poor performance. (c) The **Prediction Variance** quantification method cannot identify in-distribution data, and also suffers from the distributional shift problem.

E. Formal Definition of Minimal Distance Between Episode and Dataset

In Section 5.1 we demonstrate the minimal distance between episodes and datasets. To give a formal definition of this distance measure, we first transform an episode τ of length H to a vector v^τ as:

$$v^\tau = \langle s_0, a_0, r_0, s_1, a_1, r_1, ..., s_{H-1}, a_{H-1}, r_{H-1} \rangle. \tag{204}$$

v^τ contains information about τ's reward and transition. Then we define the normalized distance $d(\tau_1, \tau_2)$ between episode τ_1 and episode τ_2 as:

$$d(\tau_1, \tau_2) = \frac{|v^{\tau_1} - v^{\tau_2}|_2}{|v^{\tau_2}|_2}, \tag{205}$$

where $|\cdot|_2$ is the L2 distance. $d(\tau_1, \tau_2)$ measures distance between τ_1 and τ_2 normalized by v^{τ_2}. Finally, the minimal distance $d_{min}(\tau, B)$ between episode τ and dataset $\mathcal{D}$ is defined as:

$$d_{min}(\tau, \mathcal{D}) = \min_{\tau' \in \mathcal{D}} d(\tau, \tau'). \tag{206}$$

d_{min} is calculated by finding the episode τ' in $\mathcal{D}$ that has the minimal normalized distance to τ.

F. Additional Experiment Results

F.1. Additional Visualization Results

Figure 8, 9 and 10 show how the three quantifications behave on the tasks evaluated in Section 5.1. Results demonstrate that the **Return-based** quantification achieves the best performance on identifying in-distribution episodes, which enables IDAQ to achieve superior adaptation performance. The **Prediction Error** quantification can identify in-distribution episodes in simple tasks like Reach-V2, Drawer-Close-V2, and Point-Robot, but fails in more complex tasks. The **Prediction Variance** quantification fails to identify in-distribution episodes in these eight tasks. These results correspond to the performance demonstrated in Table 1.

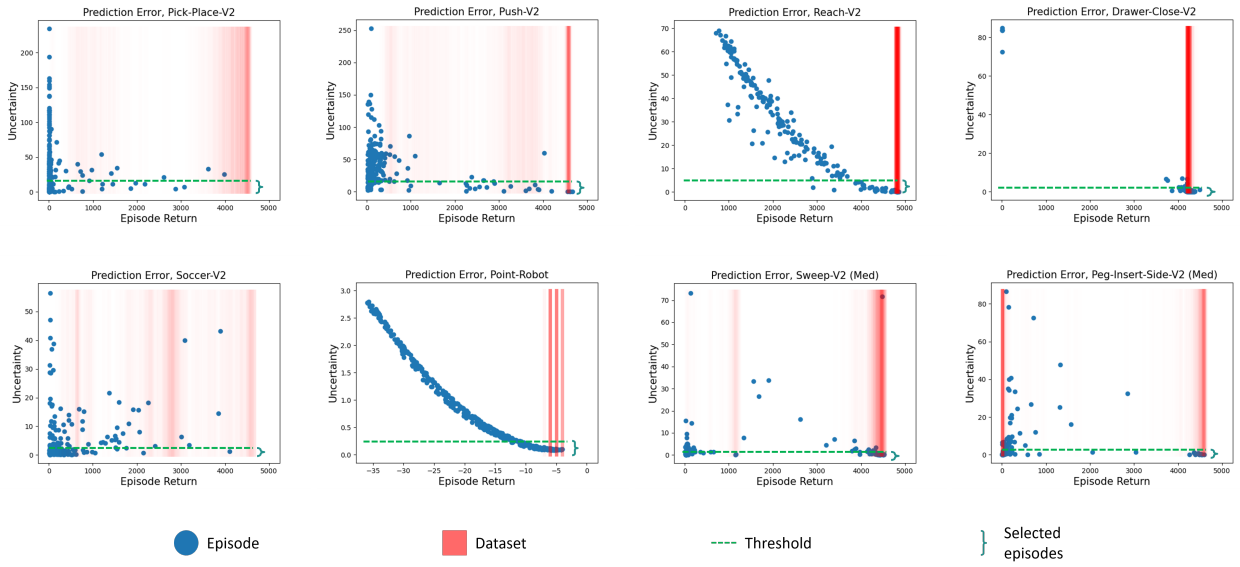


Figure 8. Visualization of the **Prediction Error** quantification's behavior on various tasks. It cannot find a good reference threshold to distinguish in-distribution episodes.

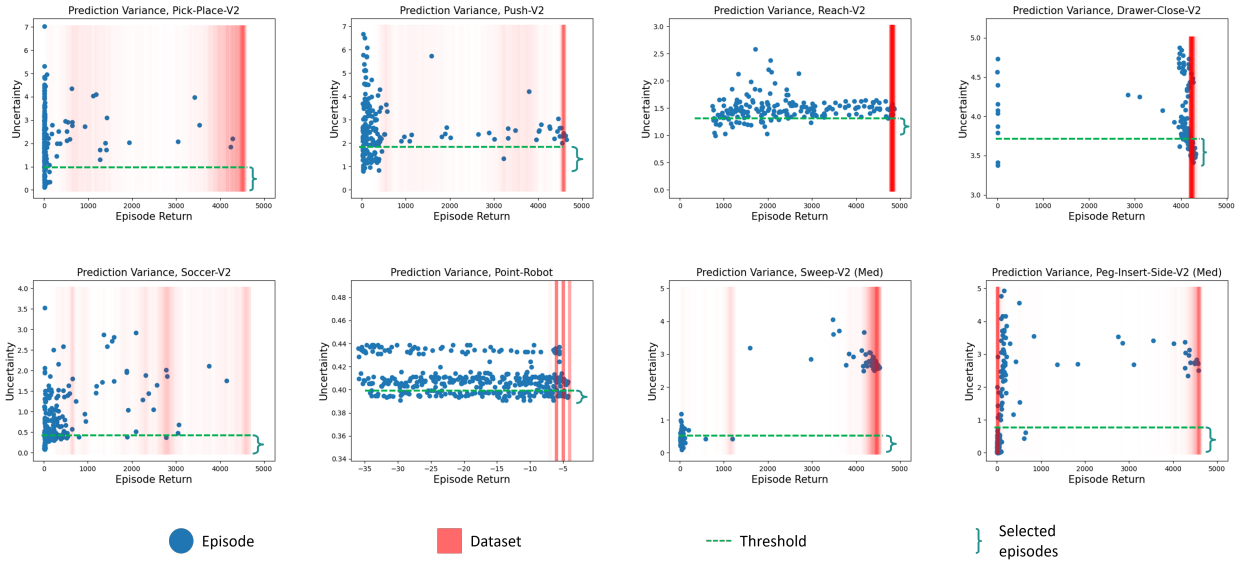


Figure 9. Visualization of the **Prediction Variance** quantification's behavior on various tasks. It may suffer from situations with higher prediction error and lower prediction variance in the medium or expert datasets.

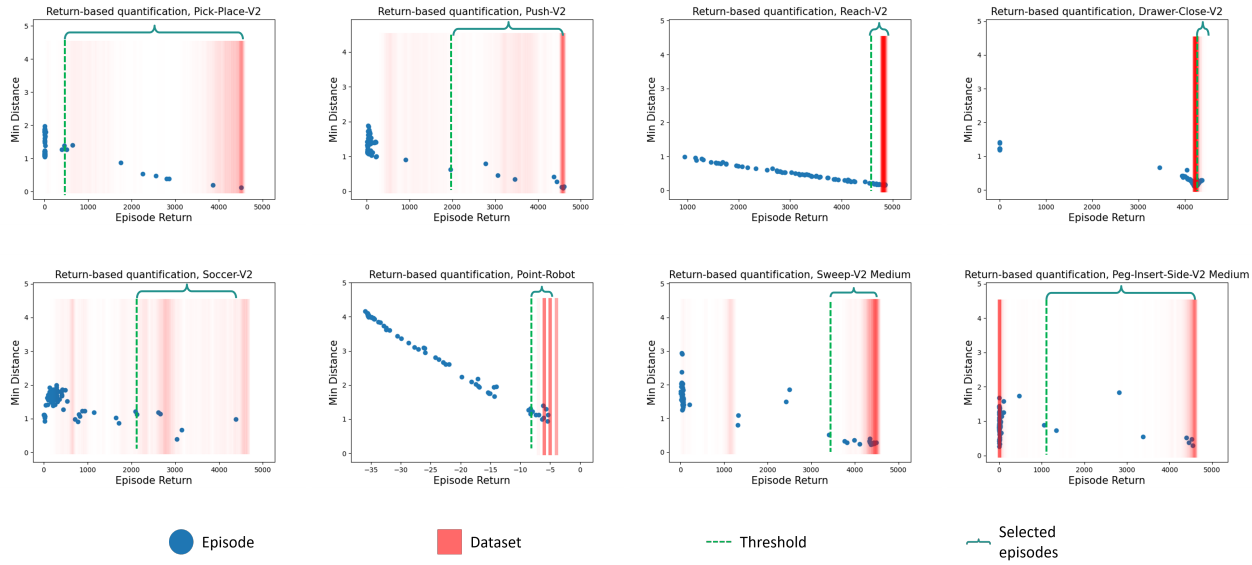


Figure 10. Visualization of the **Return-based** quantification's behavior on various tasks. It can successfully identify in-distribution episodes.

F.2. Detailed Algorithm Performance on All Tasks

Table 6, Table 7, Table 8 and Table 9 show baselines' online adaptation and offline performance on all 50 Meta-World ML1 task sets and MuJoCo tasks, respectively. IDAQ significantly outperforms baselines with online adaptation, and achieves better or comparable performance to baselines with offline adaptation.

Table 6. Comparison between FOCAL, MACAW, and BOReL with online adaptation and IDAQ on Meta-World.

Environment	IDAQ	FOCAL	MACAW	BOReL
Coffee-Push	**1.26** ± 0.13	0.66 ± 0.07	0.01 ± 0.01	0.00 ± 0.00
Faucet-Close	**1.12** ± 0.01	1.06 ± 0.02	0.07 ± 0.01	0.13 ± 0.03
Door-Unlock	**1.11** ± 0.02	0.97 ± 0.03	0.11 ± 0.01	0.13 ± 0.03
Plate-Slide-Side	**1.07** ± 0.08	0.70 ± 0.14	0.00 ± 0.00	0.00 ± 0.00
Faucet-Open	**1.05** ± 0.02	1.01 ± 0.02	0.08 ± 0.04	0.12 ± 0.05
Button-Press-Wall	**1.04** ± 0.04	0.99 ± 0.06	0.02 ± 0.00	0.01 ± 0.00
Plate-Slide	**1.01** ± 0.03	0.83 ± 0.09	0.01 ± 0.00	0.01 ± 0.00
Door-Close	**0.99** ± 0.00	0.97 ± 0.01	0.00 ± 0.00	0.37 ± 0.19
Drawer-Close	**0.99** ± 0.02	**0.96** ± 0.04	0.53 ± 0.50	0.00 ± 0.00
Plate-Slide-Back-Side	**0.97** ± 0.02	0.77 ± 0.20	0.02 ± 0.01	0.01 ± 0.00
Door-Lock	**0.97** ± 0.01	0.90 ± 0.02	0.25 ± 0.11	0.14 ± 0.00
Window-Open	**0.96** ± 0.02	0.81 ± 0.07	0.15 ± 0.11	0.03 ± 0.00
Door-Open	**0.96** ± 0.02	0.78 ± 0.13	0.06 ± 0.01	0.12 ± 0.01
Plate-Slide-Back	**0.96** ± 0.02	0.58 ± 0.06	0.21 ± 0.17	0.01 ± 0.00
Window-Close	**0.94** ± 0.01	0.79 ± 0.01	0.54 ± 0.44	0.03 ± 0.00
Reach-Wall	**0.93** ± 0.05	0.53 ± 0.18	0.82 ± 0.02	0.06 ± 0.00
Dial-Turn	**0.91** ± 0.05	0.84 ± 0.09	0.00 ± 0.00	0.00 ± 0.00
Handle-Press-Side	**0.91** ± 0.02	0.79 ± 0.10	0.51 ± 0.41	0.02 ± 0.01
Handle-Pull	**0.90** ± 0.02	0.67 ± 0.03	0.00 ± 0.00	0.00 ± 0.00
Handle-Press	**0.88** ± 0.05	**0.87** ± 0.02	0.28 ± 0.10	0.01 ± 0.00
Reach	**0.85** ± 0.03	0.62 ± 0.05	0.63 ± 0.04	0.04 ± 0.01
Lever-Pull	**0.85** ± 0.02	0.73 ± 0.07	0.20 ± 0.16	0.05 ± 0.00
Hammer	**0.84** ± 0.06	0.59 ± 0.07	0.10 ± 0.01	0.09 ± 0.01
Drawer-Open	**0.82** ± 0.06	0.64 ± 0.10	0.11 ± 0.02	0.10 ± 0.00
Sweep	**0.77** ± 0.04	0.32 ± 0.08	0.20 ± 0.20	0.00 ± 0.00
Button-Press	**0.74** ± 0.08	**0.68** ± 0.14	0.02 ± 0.01	0.01 ± 0.01
Stick-Push	**0.73** ± 0.09	0.46 ± 0.15	0.17 ± 0.17	0.00 ± 0.00
Coffee-Button	**0.73** ± 0.14	**0.66** ± 0.16	0.15 ± 0.13	0.02 ± 0.00
Push-Wall	**0.71** ± 0.15	0.43 ± 0.06	0.23 ± 0.18	0.00 ± 0.00
Shelf-Place	**0.70** ± 0.18	0.32 ± 0.11	0.01 ± 0.01	0.00 ± 0.00
Basketball	**0.64** ± 0.15	0.41 ± 0.24	0.00 ± 0.00	0.00 ± 0.00
Hand-Insert	**0.63** ± 0.04	0.29 ± 0.07	0.02 ± 0.01	0.00 ± 0.00
Sweep-Into	**0.61** ± 0.06	0.33 ± 0.05	0.00 ± 0.00	0.01 ± 0.00
Button-Press-Topdown	**0.57** ± 0.11	0.45 ± 0.06	0.38 ± 0.36	0.02 ± 0.02
Peg-Unplug-Side	**0.56** ± 0.07	0.19 ± 0.09	0.00 ± 0.00	0.00 ± 0.00
Assembly	**0.55** ± 0.13	0.28 ± 0.05	0.33 ± 0.01	0.04 ± 0.00
Push	**0.55** ± 0.10	0.34 ± 0.14	0.28 ± 0.19	0.00 ± 0.00
Bin-Picking	0.53 ± 0.16	0.31 ± 0.21	**0.66** ± 0.11	0.00 ± 0.00
Push-Back	**0.52** ± 0.05	0.16 ± 0.04	0.00 ± 0.00	0.00 ± 0.00
Box-Close	**0.51** ± 0.11	0.15 ± 0.09	0.36 ± 0.11	0.05 ± 0.01
Soccer	**0.44** ± 0.04	0.11 ± 0.03	**0.38** ± 0.31	0.04 ± 0.02
Button-Press-Topdown-Wall	**0.43** ± 0.03	**0.40** ± 0.07	0.05 ± 0.02	0.05 ± 0.01
Disassemble	**0.42** ± 0.14	0.26 ± 0.04	0.05 ± 0.00	0.04 ± 0.00
Coffee-Pull	**0.40** ± 0.05	0.23 ± 0.04	0.19 ± 0.12	0.00 ± 0.00
Stick-Pull	**0.32** ± 0.06	0.17 ± 0.07	0.00 ± 0.00	0.00 ± 0.00
Peg-Insert-Side	**0.30** ± 0.04	0.08 ± 0.03	0.00 ± 0.00	0.00 ± 0.00
Pick-Place-Wall	0.28 ± 0.12	0.09 ± 0.04	**0.39** ± 0.25	0.00 ± 0.00
Pick-Out-Of-Hole	0.26 ± 0.25	0.16 ± 0.16	**0.59** ± 0.06	0.00 ± 0.00
Pick-Place	**0.20** ± 0.03	0.07 ± 0.02	0.05 ± 0.05	0.00 ± 0.00
Handle-Pull-Side	**0.14** ± 0.04	**0.13** ± 0.09	0.00 ± 0.00	0.00 ± 0.00
Average	**0.73** ± 0.07	0.53 ± 0.08	0.18 ± 0.09	0.04 ± 0.01

Table 7. Comparison between baselines with offline adaptation and IDAQ on Meta-World tasks. "-V2" is omitted for brevity.

Environment	IDAQ	FOCAL with Expert Context	MACAW with Expert Context
Coffee-Push	**1.26** ± 0.13	0.50 ± 0.06	**1.14** ± 0.27
Faucet-Close	**1.12** ± 0.01	1.07 ± 0.02	1.01 ± 0.01
Door-Unlock	**1.11** ± 0.02	0.96 ± 0.03	0.99 ± 0.04
Plate-Slide-Side	**1.07** ± 0.08	0.75 ± 0.09	0.91 ± 0.09
Faucet-Open	**1.05** ± 0.02	**1.00** ± 0.02	0.99 ± 0.01
Button-Press-Wall	**1.04** ± 0.04	0.98 ± 0.05	**0.99** ± 0.01
Plate-Slide	**1.01** ± 0.03	**1.00** ± 0.03	0.67 ± 0.07
Door-Close	**0.99** ± 0.00	0.97 ± 0.01	0.92 ± 0.05
Drawer-Close	**0.99** ± 0.02	0.96 ± 0.04	**1.00** ± 0.01
Plate-Slide-Back-Side	**0.97** ± 0.02	0.90 ± 0.07	0.80 ± 0.05
Door-Lock	**0.97** ± 0.01	0.88 ± 0.04	**0.97** ± 0.03
Window-Open	**0.96** ± 0.02	**0.93** ± 0.05	**0.98** ± 0.02
Door-Open	**0.96** ± 0.02	0.90 ± 0.02	**0.99** ± 0.02
Plate-Slide-Back	**0.96** ± 0.02	**0.93** ± 0.01	0.55 ± 0.11
Window-Close	0.94 ± 0.01	0.73 ± 0.02	**1.00** ± 0.01
Reach-Wall	**0.93** ± 0.05	**0.91** ± 0.05	0.82 ± 0.02
Dial-Turn	0.91 ± 0.05	0.84 ± 0.08	**0.98** ± 0.01
Handle-Press-Side	**0.91** ± 0.02	0.87 ± 0.04	0.82 ± 0.10
Handle-Pull	**0.90** ± 0.02	0.70 ± 0.05	**0.95** ± 0.05
Handle-Press	**0.88** ± 0.05	**0.79** ± 0.08	0.56 ± 0.19
Reach	**0.85** ± 0.03	**0.83** ± 0.05	0.64 ± 0.08
Lever-Pull	0.85 ± 0.02	0.76 ± 0.03	**0.97** ± 0.07
Hammer	**0.84** ± 0.06	0.78 ± 0.04	0.40 ± 0.18
Drawer-Open	0.82 ± 0.06	0.73 ± 0.11	**0.98** ± 0.01
Sweep	0.77 ± 0.04	0.74 ± 0.02	**0.98** ± 0.01
Button-Press	**0.74** ± 0.08	0.63 ± 0.09	**0.71** ± 0.04
Stick-Push	**0.73** ± 0.09	0.14 ± 0.09	**0.67** ± 0.09
Coffee-Button	**0.73** ± 0.14	**0.61** ± 0.20	0.21 ± 0.11
Push-Wall	0.71 ± 0.15	**0.90** ± 0.12	**0.96** ± 0.09
Shelf-Place	**0.70** ± 0.18	0.57 ± 0.13	0.55 ± 0.03
Basketball	**0.64** ± 0.15	0.49 ± 0.17	0.47 ± 0.18
Hand-Insert	**0.63** ± 0.04	**0.64** ± 0.09	0.20 ± 0.09
Sweep-Into	**0.61** ± 0.06	**0.64** ± 0.09	0.00 ± 0.00
Button-Press-Topdown	0.57 ± 0.11	0.48 ± 0.11	**0.92** ± 0.04
Peg-Unplug-Side	**0.56** ± 0.07	**0.57** ± 0.10	0.18 ± 0.10
Assembly	0.55 ± 0.13	**0.64** ± 0.03	0.36 ± 0.02
Push	0.55 ± 0.10	**0.98** ± 0.13	0.86 ± 0.02
Bin-Picking	0.53 ± 0.16	**0.61** ± 0.12	**0.63** ± 0.11
Push-Back	**0.52** ± 0.05	**0.52** ± 0.16	0.15 ± 0.09
Box-Close	**0.51** ± 0.11	**0.56** ± 0.08	0.35 ± 0.11
Soccer	0.44 ± 0.04	0.45 ± 0.03	**0.59** ± 0.11
Button-Press-Topdown-Wall	**0.43** ± 0.03	**0.40** ± 0.06	**0.43** ± 0.06
Disassemble	0.42 ± 0.14	0.23 ± 0.05	**0.50** ± 0.06
Coffee-Pull	0.40 ± 0.05	**0.58** ± 0.11	0.45 ± 0.11
Stick-Pull	**0.32** ± 0.06	0.18 ± 0.06	**0.27** ± 0.09
Peg-Insert-Side	0.30 ± 0.04	**0.52** ± 0.08	0.25 ± 0.04
Pick-Place-Wall	**0.28** ± 0.12	0.13 ± 0.07	**0.21** ± 0.16
Pick-Out-Of-Hole	0.26 ± 0.25	0.27 ± 0.27	**0.59** ± 0.08
Pick-Place	0.20 ± 0.03	0.29 ± 0.11	**0.72** ± 0.09
Handle-Pull-Side	0.14 ± 0.04	0.09 ± 0.05	**0.94** ± 0.08
Average	**0.73** ± 0.07	0.67 ± 0.07	0.68 ± 0.07

Table 8. Comparison between FOCAL, MACAW, and BOReL with online adaptation and IDAQ on MuJoCo tasks.

Environment	IDAQ	FOCAL	MACAW	BOReL
Cheetah-Vel	**-171.52** ± 21.96	-287.70 ± 30.62	-233.97 ± 23.46	-301.4 ± 36.8
Point-Robot	**-5.10** ± 0.26	-15.38 ± 0.95	-14.61 ± 0.98	-17.28 ± 1.16
Point-Robot-Sparse	**7.78** ± 0.64	0.83 ± 0.37	0.00 ± 0.00	0.00 ± 0.00

Table 9. Comparison between baselines with offline adaptation and IDAQ on MuJoCo tasks.

Environment	IDAQ	FOCAL with Expert Context	MACAW with Expert Context
Cheetah-Vel	**-171.52** ± 21.96	**-156.07** ± 23.22	-292.92 ± 36.66
Point-Robot	**-5.10** ± 0.26	**-4.68** ± 0.18	-19.60 ± 1.15
Point-Robot-Sparse	**7.78** ± 0.64	**8.37** ± 0.67	0.00 ± 0.00

F.3. Dataset Returns

Table 10 and Table 11 show the average returns of the offline datasets used in meta-training.

Table 10. Dataset average returns on 50 Meta-World tasks.

Environment	Dataset Return
Coffee-Push	1487.84
Faucet-Close	4039.52
Door-Unlock	3653.40
Plate-Slide-Side	3530.06
Faucet-Open	4145.18
Button-Press-Wall	2899.34
Plate-Slide	4395.86
Door-Close	4519.77
Drawer-Close	4233.04
Plate-Slide-Back-Side	4735.74
Door-Lock	3352.35
Window-Open	4382.87
Door-Open	4401.24
Plate-Slide-Back	4732.90
Window-Close	3572.99
Reach-Wall	4805.38
Dial-Turn	3824.44
Handle-Press-Side	4836.04
Handle-Pull	3878.46
Handle-Press	4851.51
Reach	4820.84
Lever-Pull	914.01
Hammer	4251.57
Drawer-Open	4041.54
Sweep	4354.57
Button	3855.10
Stick-Push	4230.52
Coffee-Button	4049.44
Push-Wall	3699.66
Shelf-Place	2813.55
Basketball	4071.31
Hand-Insert	3963.11
Sweep-Into	4019.97
Button-Press-Topdown	3570.22
Peg-Unplug-Side	4235.23
Assembly	4238.57
Push	3147.80
Bin-Picking	4257.54
Push-Back	3841.90
Box-Close	4012.39
Soccer	2798.96
Button-Press-Topdown-Wall	3781.99
Disassemble	3865.37
Coffee-Pull	4166.44
Stick-Pull	4015.26
Peg-Insert-Side	3826.91
Pick-Place-Wall	2499.35
Pick-Out-Of-Hole	3405.74
Pick-Place	3482.43
Handle-Pull-Side	2425.14

Table 11. Dataset average returns on Meta-World-Meidum tasks and MuJoCo tasks.

Environment	Dataset Return
Sweep (Med)	2874.00
Peg-Insert-Side (Med)	2342.34
Point-Robot	-5.72
Point-Robot-Sparse	7.28
Cheetah-Vel	-138.29

G. Additional Ablation Studies

In this section, we will further conduct various ablation studies to investigate the robustness of IDAQ in dataset quality and hyper-parameters.

Reference stage length. Table 12 shows IDAQ's performance with different reference stage lengths. The total number of adaptation episodes is 20. We find that IDAQ performs well during 10-15 episodes, which is 50%-75% of the total number of adaptation episodes. A small reference stage length (5) may lead to a possibly unreliable task belief and cause a degradation in performance. The 19-episode does not perform the iterative optimization process, and the task belief updates will not converge.

Dataset Quality. We test IDAQ and baselines with several "medium" datasets, which are collected by periodically evaluating policies of SAC. As shown in Table 13, IDAQ still significantly outperforms baseline algorithms on medium datasets, which implies IDAQ's ability to learn various datasets.

Table 12. IDAQ's performance with various initial stage lengths.

Environment	5 Episodes	10 Episodes	15 Episodes	19 Episodes
Point-Robot	-6.04 $\pm$ 0.31	**-5.11** $\pm$ 0.21	**-5.10** $\pm$ 0.26	-5.37 $\pm$ 0.11
Point-Robot-Sparse	4.04 $\pm$ 0.58	**7.78** $\pm$ 0.64	**8.07** $\pm$ 0.62	7.29 $\pm$ 0.50

Table 13. Algorithms' performance on datasets of various qualities.

Environment	IDAQ	FOCAL	MACAW
Sweep	**0.77** $\pm$ 0.04	0.32 $\pm$ 0.08	0.20 $\pm$ 0.20
Sweep (Med)	**0.59** $\pm$ 0.13	0.38 $\pm$ 0.13	0.04 $\pm$ 0.03
Peg-Insert-Side	**0.30** $\pm$ 0.04	0.08 $\pm$ 0.03	0.00 $\pm$ 0.00
Peg-Insert-Side (Med)	**0.30** $\pm$ 0.14	0.10 $\pm$ 0.07	0.00 $\pm$ 0.00

Hyper-parameters. The uncertainty quantifications are robust to the hyperparameters in this paper. Prior work (Lu et al., 2021) introduces three sensitive hyperparameters: penalty scale, rollout length, and the number of models in offline model-based RL. However, in the in-distribution online adaptation framework of IDAQ, penalty scale and rollout length are not required. IDAQ utilizes uncertainty quantifications to rank online episodes for deriving in-distribution contexts instead of using it as a regularization. For IDAQ+Prediction Error and IDAQ+Prediction Variance, the hyperparameter of these uncertainty quantifications is the number of models L in an ensemble. We perform ablation studies on this hyperparameter.

Results in Table 14 and Table 15 show that IDAQ+Prediction Error and IDAQ+Prediction Variance are robust to the choice of the number of models in an ensemble, and IDAQ+Return significantly outperforms IDAQ+Prediction Error and IDAQ+Prediction Variance with different hyperparameter values.

Other baselines. Our approach IDAQ is compatible with existing context-based offline meta-RL methods. FOCAL is one of such state-of-the-art methods that can address the challenge of *MDP Ambiguity* (Dorfman et al., 2021) in offline meta-training. Due to the task-dependent behavior policy, the sub-datasets of different tasks may be disjoint and a promising

Table 14. Ablation on the number of ensembles L in IDAQ+Prediction Error.

Example Env	$L = 2$	$L = 4$	$L = 8$	$L = 12$	IDAQ+Return
Sweep (Med)	0.13 ± 0.03	0.14 ± 0.05	0.16 ± 0.03	0.17 ± 0.04	0.59 ± 0.13
Point-Robot	-6.33 ± 0.22	-5.70 ± 0.05	-6.14 ± 0.24	-5.98 ± 0.24	-5.10 ± 0.26
Reach	0.85 ± 0.02	0.86 ± 0.02	0.85 ± 0.02	0.86 ± 0.01	0.85 ± 0.03

Table 15. Ablation on the number of ensembles L in IDAQ+Prediction Variance.

Example Env	$L = 2$	$L = 4$	$L = 8$	$L = 12$	IDAQ+Return
Sweep (Med)	0.05 ± 0.02	0.05 ± 0.02	0.06 ± 0.02	0.05 ± 0.02	0.59 ± 0.13
Point-Robot	-19.30 ± 0.92	-21.29 ± 0.85	-20.51 ± 1.06	-19.48 ± 0.93	-5.10 ± 0.26
Reach	0.47 ± 0.04	0.46 ± 0.03	0.47 ± 0.03	0.46 ± 0.03	0.85 ± 0.03

solution is to apply contrastive loss (Li et al., 2020b; Yuan & Lu, 2022) on the latent task embeddings to distinguish tasks. Offline PEARL+Contrastive Loss (abbreviated as "OP+CL") is another popular offline meta-training baseline in the literature (Li et al., 2020b; Yuan & Lu, 2022). Combining IDAQ with OP+CL, we will investigate how much improvement can be achieved by in-distribution online adaptation in OP+CL. Results in Table 16 demonstrate that IDAQ can significantly improve over OP+CL (the "OP+CL+IDAQ" column), and achieves similar or slightly lower performance than FOCAL+IDAQ.

Table 16. Improvement of OP+CL+IDAQ over the OP+CL baseline.

Example Env	FOCAL+IDAQ	FOCAL	OP+CL+IDAQ	OP+CL
Push	**0.55** ± 0.10	0.34 ± 0.14	0.45 ± 0.04	0.19 ± 0.02
Pick-Place	**0.20** ± 0.03	0.07 ± 0.02	**0.19** ± 0.04	0.08 ± 0.02
Soccer	**0.44** ± 0.04	0.11 ± 0.03	**0.42** ± 0.05	0.19 ± 0.02
Drawer-Close	**0.99** ± 0.02	**0.96** ± 0.04	**1.00** ± 0.01	0.82 ± 0.04
Reach	0.85 ± 0.03	0.62 ± 0.05	**0.90** ± 0.01	0.59 ± 0.03
Sweep (Med)	**0.59** ± 0.13	0.38 ± 0.13	0.48 ± 0.09	0.19 ± 0.06
Peg-Insert-Side (Med)	**0.30** ± 0.14	0.10 ± 0.07	0.19 ± 0.04	0.06 ± 0.02