A Bayesian Fast-Slow Framework to Mitigate Interference in Non-Stationary Reinforcement Learning

Yihuan Mao
Institute for Interdisciplinary Information Sciences
Tsinghua University
maoyh1024@gmail.com

Chongjie Zhang
Department of Computer Science & Engineering
Washington University in St. Louis
chongjie@wustl.edu

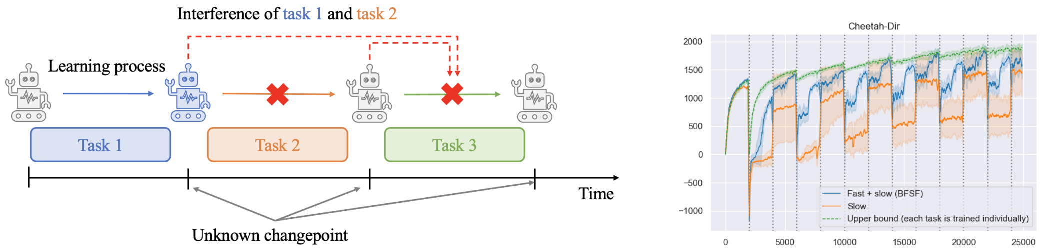
Abstract

Given the ever-changing nature of the world and its inhabitants, agents must possess the ability to adapt and evolve over time. Recent research in Given the ever-changing nature of the world and its inhabitants, agents must possess the ability to adapt and evolve over time. Recent research in non-stationary MDPs has focused on addressing this challenge, providing algorithms inspired by task inference techniques. However, these methods ignore the detrimental effects of interference, which particularly harm performance in contradictory tasks, leading to low efficiency in some environments. To address this issue, we propose a Bayesian Fast-Slow Framework (BFSF) that tackles both cross-task generalization and resistance to cross-task interference. Our framework consists of two components: a 'fast' policy, learned from recent data, and a 'slow' policy, learned through meta-reinforcement learning (meta-RL) using data from all previous tasks. A Bayesian estimation mechanism determines the current choice of 'fast' or 'slow' policy, balancing exploration and exploitation. Additionally, in the 'fast' policy, we introduce a dual-reset mechanism and a data relabeling technique to further accelerate convergence when encountering new tasks. Experiments demonstrate that our algorithm effectively mitigates interference and outperforms baseline approaches. Code is available at `https://github.com/cedesu/BFSF`.

Reinforcement Learning (RL) in non-stationary environments has long attracted significant attention, leading to the emergence of research areas such as continual RL [30, 19] and non-stationary MDPs [7, 24, 31]. These areas approach challenges from different perspectives. For example, catastrophic forgetting, a well-known issue in continual RL, arises due to limited memory storage. In contrast, research on non-stationary MDPs focuses on understanding the underlying dynamics of the environment to facilitate better adaptation across varying contexts.

One critical challenge in non-stationary environments is interference, where the learning process is negatively impacted by experiences from previous tasks, as illustrated in Figure 1. This interference arises primarily because task boundaries are either unknown or absent in the streaming task setting. Many real-world problems exhibit such non-stationarity and suffer from interference. For instance, a UAV must adapt its behavior under varying weather conditions [27, 26]. Similarly, the evolving regime of the stock market can be viewed as a time-varying environment, where an effective strategy

39th Conference on Neural Information Processing Systems (NeurIPS 2025).



(a) The diagram of interference in non-stationary MDPs.

(b) The learning curve of how BFSF alleviates the problem of interference.

Figure 1: Figure 1(a) illustrates the interference problem in non-stationary MDPs. The agent learns to perform well on the current task, but the changepoint is unknown. As a result, when the agent begins learning a new task (e.g., task 2), experience from previous tasks (e.g., task 1) can hinder performance. This interference phenomenon also occurs across consecutive tasks. To address this, we propose BFSF, which incorporates a 'fast' policy that learns from recent data to mitigate interference, alongside a 'slow' policy using meta-RL to learn a context-based policy from all previous tasks. Figure 1(b) demonstrates BFSF's ability to resist interference in the Cheetah-Dir task, which involves two contradictory tasks: moving forward and backward. In the second phase, the learning curve shows less disruption compared to the 'slow' policy only. The highest performance throughout the non-stationary MDP process is close to the upper bound, which represents the scenario where the two tasks are trained separately.

must stay adaptive while leveraging past experience [13, 3]. These real-world scenarios underscore the urgent need for a framework that can operate efficiently in non-stationary MDPs with interference.

Despite its significant negative impact on performance, the issue of interference has been largely overlooked in the literature. Some existing works [18] address interference from the perspective of representation, while others [20] discuss the inverse interference of current tasks on previously learned tasks, a phenomenon referred to as catastrophic forgetting in continual RL. This lack of attention to cross-task interference is concerning, as it can severely degrade performance in successive tasks. In this work, we specifically analyze the effects of interference and propose effective strategies to mitigate its detrimental impacts.

To address the interference problem, we introduce the Bayesian Fast-Slow Framework (BFSF). This framework dynamically selects between two learning strategies: a 'fast' policy, which quickly adapts to new tasks using recent data, and a 'slow' policy, which is learned through meta-RL and captures knowledge from historical data. Unlike previous approaches that focus solely on latter, often leading to severe interference in the face of sudden task changes, our framework not only mitigates interference but also preserves the advantages of meta-RL. A Bayesian estimation mechanism is employed in each epoch to decide which policy, fast or slow, is more promising based on recent history. Only the most recent returns are used to update the Bayesian estimates, ensuring that outdated data does not influence the decision.

We also identify that the 'fast' policy can sometimes underperform. One reason is that neural networks often experience performance degradation when trained on data from different distributions, a common issue in non-stationary environments. To address this, we introduce a dual-reset mechanism that periodically reinitializes one of the dual networks to prevent degradation, while alternating between the networks to ensure stable performance. Another challenge is that learning from scratch typically requires extensive online interaction. To mitigate this, we propose data relabeling, utilizing historical data from previous tasks to enhance learning efficiency and improve performance in few-shot settings when facing new tasks.

In summary, our contributions are twofold: i) We analyze the interference problem in non-stationary MDPs. ii) We propose the Bayesian Fast-Slow Framework (BFSF), which combines a fast policy, enhanced by a dual-reset mechanism and data relabeling, to efficiently handle recent tasks, and a slow policy for cross-task generalization. Through Bayesian estimation, we effectively address interference and improve overall performance. Experimental results demonstrate BFSF's superiority in resisting interference and outperforming baseline methods across various non-stationary environments.

1 Preliminaries

Notations and problem definition A Markov Decision Process (MDP) is defined as $M = \langle S, A, P, R \rangle$, where S and A represent the state and action spaces, respectively. The transition function of the environment is denoted as P, and R represents the reward function. The expected return of a policy is given by $\mathbb{E}[\sum_{t=0}^{\infty} R_t]$. In non-stationary MDPs, the underlying MDP evolves over time. These changes can occur sequentially, such as $M_1, M_2, \cdots$, or gradually over time. The objective is to maximize the expected return, $\mathbb{E}[\sum_{t=0}^{\infty} R_t]$ with the evolving dynamics of the MDP.

Context-based policy In a standard MDP, the policy function is defined as $\pi(a|s)$, which determines the probability of selecting action a given state s. In non-stationary settings, a context-based policy is introduced to adapt to varying environments. This policy is denoted as $\pi(a|s, c)$, where c is the context, a set of trajectories related to the current environment. A trajectory consists of the sequence of states, actions, and rewards at each timestep, expressed as $s_0, a_0, r_0, s_1, a_1, r_1, s_2, a_2, r_2, \ldots$. In implementation, the context is collected from recent interactions, reflecting the underlying MDP.

The technique of learning the context-based policy has been extensively studied in the field of meta-reinforcement learning (meta-RL) [24, 37]. However, meta-RL differs from the non-stationary MDPs setting in this work. In meta-RL, task information is explicitly available, and there is no continuous adaptation process. Context-based meta-RL methods typically map the contextual information, often represented as transition data, into a latent space $\mathcal{Z}$. By assigning a latent variable $z \in \mathcal{Z}$ to represent the task, this approach effectively frames the problem as a partially observable MDP (POMDP) [16], where z constitutes the unobserved portion of the state. In meta RL, PEARL learns the posterior distribution $q(z|c)$, which means the posterior latent variable distribution given the context c, and uses posterior sampling to sample z to integrate these latent variables with off-policy RL algorithms.

Algorithm 1 Bayesian fast-slow framework (BFSF)

```
Input: A 'fast' policy $\pi_{fast}$ (including the dual policies $\pi_{fast}^{(1)}, \pi_{fast}^{(2)}$), a 'slow' policy $\pi_{slow}$, the
number of epochs $E$, the window of recent data $w$
Initialize return list $\{R_i\}_{i\in 1\cdots E}$ and choice list $\{\text{choice}_j\}_{i\in 1\cdots E}$
for epoch $e = 1 \cdots E$ do
    # Bayesian inference of the expected return
    $\hat{R}_{fast} = \text{Posterior}(\{R_{i\in[e-w,e]}|\text{choice}_i = fast\})$
    $\hat{R}_{slow} = \text{Posterior}(\{R_{i\in[e-w,e]}|\text{choice}_i = slow\})$
    # Online interaction
    if $\hat{R}_{fast} > \hat{R}_{slow}$ then
        $\text{choice}_e := fast$
        Collect data using $\pi_{fast}$
    else
        $\text{choice}_e := slow$
        Collect data using $\pi_{slow}$
    end if
    # Training
    Update $\pi_{fast}$ by Algorithm 2
    Update $\pi_{slow}$ by the meta-RL algorithm
end for
```

2 Bayesian Fast-Slow Framework (BFSF)

The Bayesian Fast-Slow Framework (BFSF) is designed to mitigate interference by dynamically deploying either a 'fast' policy, which learns from recent data, or a 'slow' policy, trained using meta-RL principles. The term 'fast' arises from its 'fast-adaptation' ability to learn directly and efficiently from recent data. In contrast, the 'slow' policy enables cross-task understanding and generalization, which may hinder training speed, especially when the number of observed tasks is limited in early phase. The decision is made based on Bayesian estimation of current expected return.

As illustrated in Algorithm 1, during each epoch, Bayesian inference is applied to estimate the posterior expected return using the recent return history, for both the fast and slow policies. Let R_i denote the return obtained in the i-th epoch, and choice_i indicate whether the 'fast' or 'slow' policy was selected during that epoch. During the online interaction phase, the policy with the higher estimated posterior value is selected, aiming to generate higher-quality experience. At the end of each epoch, both the fast and slow policies are updated using their respective replay buffers. The detailed computation of the Bayesian posterior, described in lines 5 and 6 of Algorithm 1, is elaborated in Section 2.1. In addition to the online interaction and Bayesian estimation, the training process for the 'fast' policy is detailed in Algorithm 2, while the 'slow' policy is trained according to the context-based meta-RL algorithm PEARL [24], which is one of the first context-based methods and serves as the baseline for numerous subsequent works. The visualization of choosing the 'fast' or 'slow' policy is provided in Appendix E.

2.1 Bayesian Inference

The detailed update rule for Bayesian posterior estimation is outlined below. For simplicity, assume that the recent returns of a given policy, $R_{i_1}, R_{i_2}, \ldots$, are approximately drawn from a normal distribution $\mathcal{N}(\mu, 1/\phi)$, where ϕ is a constant. While this assumption is commonly used, other distributional forms could also be considered depending on the context. The prior distribution for the parameter μ is assumed to follow $\mu \sim \mathcal{N}(\mu_0, 1/\phi_0)$. The posterior estimation of μ then follows the standard derivation below, as detailed in Appendix D.

$$\begin{gathered} p(\mu|\{R_{i_1}, R_{i_2}, \cdots\}, \mu_0, \phi_0) \sim \mathcal{N}(\mu_1, 1/\sigma_1^2), \\ \text{where } \mu_1 = \frac{\phi_0\mu_0 + n\phi\overline{R}}{\phi_0 + n\phi}, \sigma_1^2 = \frac{1}{\phi_0 + n\phi}. \end{gathered} \tag{1}$$

To better interpret the result of Bayesian inference, note that the posterior mean μ_1 can be decomposed:

$$\mu_1 = \frac{\phi_0\mu_0 + n\phi\overline{R}}{\phi_0 + n\phi} = \frac{\phi_0}{\phi_0 + n\phi}\mu_0 + \frac{n\phi}{\phi_0 + n\phi}\overline{R}. \tag{2}$$

It shows that the posterior mean μ_1 is a weighted average of the prior mean μ_0 and the sample average $\overline{R}$. As more samples are collected, the weight shifts toward trusting the sample average $\overline{R}$. Conversely, when only a few samples are available, the posterior relies more heavily on the prior μ_0.

2.2 'Fast' Policy Learning

The 'fast' policy, learned from recent data, is critical for mitigating interference in non-stationary MDPs. However, the standard learning paradigm often encounters challenges under these conditions. To address these issues, we propose specific structural designs that significantly enhance the efficiency and adaptability of the 'fast' policy.

Algorithm 2 Training process of the 'fast' policy.

```
Input: The 'fast' policy $\pi_{fast}$ (including the dual policies $\pi_{fast}^{(1)}, \pi_{fast}^{(2)}$), a contextual dynamics model $M$, current epoch $e$, reset frequency $\nu$
Output: The updated $\pi_{fast}$
# Dual-reset mechanism
**if** $e \bmod \nu = 0$ **then**
    $\pi_{fast}^{(1)}, \pi_{fast}^{(2)} = \pi_{fast}^{(2)}, \mathrm{Init}(\pi_{fast}^{(1)})$
**end if**
# Data relabeling
Relabel the recent data by $M$ using the recent trajectories as the context.
# Training process
Train $\pi_{fast}^{(1)}, \pi_{fast}^{(2)}$ using the relabeled data
```

Dual-reset Mechanism A key challenge in continual learning is the performance degradation of neural networks when trained on successive tasks. One common observation is that the learning curve for the second task often struggles to converge to an optimal point, even when task difficulty is comparable (as detailed in Appendix C.2). This phenomenon is also noted and studied in ITER [14].

To address this, we propose the dual-reset mechanism, as outlined in Algorithm 2, which mitigates performance degradation by periodically reinitializing the model. However, to avoid the inferior performance typically observed immediately after reinitialization, we introduce a dual-model system. This ensures that during any interaction phase, even directly after initialization, a fully trained model is always available for deployment.

Data Relabeling Another challenge lies in the limited recent data available for the 'fast' policy, which is necessary for rapid adaptation in non-stationary environments. This small dataset size may not support learning a robust policy, especially over prolonged training periods, in contrast to the 'slow' policy that can utilize the entire historical dataset. As a result, the 'fast' policy tends to perform significantly worse, as shown in Figure 5. To address this limitation, we incorporate data relabeling, which significantly enhances the amount of usable data, enabling the learning of a stronger policy.

Specifically, the data relabeling process relies on maintaining a context-based dynamics model. This model takes the context, state, and action as input, and outputs the relabeled next state and reward. Leveraging the context-based property, it becomes possible to relabel historical data from other tasks into the context of the current task. By combining relabeled data with the original dataset, the learning efficiency of the 'fast' policy is substantially improved. Further details can be found in Appendix C.1.

2.3 Theoretical Analysis

In this section, we provide a theoretical analysis of the sub-optimality bound of the Bayesian Fast-Slow Framework (BFSF) and present Theorem 2.1.

Theorem 2.1.

$$\begin{aligned}&\text{Suboptimality}(BFSF)\\ \leq&[\mathcal{D}_{\ell_1}(p_M, p_{M'})(r_{max}+V_{max})+r_{diff}]H\\ +&|U_{r,r'}(\pi^*_{M'})|+\frac{1}{2}V_{max}\mathcal{D}_{\ell_1}(p_{M'}(s,a),p_M(s,a)),\end{aligned} \tag{3}$$

where M, M' denote the original and relabeled MDPs, and $p_M, p_{M'}$ are their transition functions. H is the horizon, r_{max}, V_{max} are the maximum reward and value, r_{diff} represents the maximum reward gap between M, M', $U_{r,r'}(\pi)$ is defined as $\mathbb{E}_{(s,a)\sim\rho^\pi_M}[r'(s,a)-r(s,a)]$, and $\mathcal{D}_{\ell_1}$ is the L1 distance.

The following provides a proof sketch and interpretation of Theorem 2.1. First, the suboptimality is defined as the minimum between the fast and slow policies, with the Bayesian estimation serving as an unbiased estimate. We focus on the suboptimality of the 'fast' policy, $|\eta_M(\pi^*_M)-\eta_M(\pi^*_{M'})|$. This suboptimality can then be decomposed into several components as detailed in Appendix A.1. For ease of comprehension, the first term represents the gap in optimal expected return between the relabeled and original MDPs. The second and third terms arise from the performance difference of the same policy under different dynamics. A further analysis of the bound that incorporates optimization error is provided in Appendix A.2.

3 Experiments

The Bayesian Fast-Slow Framework (BFSF) is designed to address the complexities of non-stationary MDPs, specifically tackling the interference issue while ensuring generalization across experiences from different tasks. In this section, we focus on two main questions: i) How does BFSF overcome interference in non-stationary environments? ii) How does BFSF perform in real-world scenarios? iii) How do the individual components of BFSF contribute to improving performance?

To answer the first question, we provide experimental comparisons with baselines in Section 3.1, demonstrating the superiority of BFSF in mitigating interference. For the second question, we design an infinite-world simulation to better approximate real-world conditions and evaluate the performance

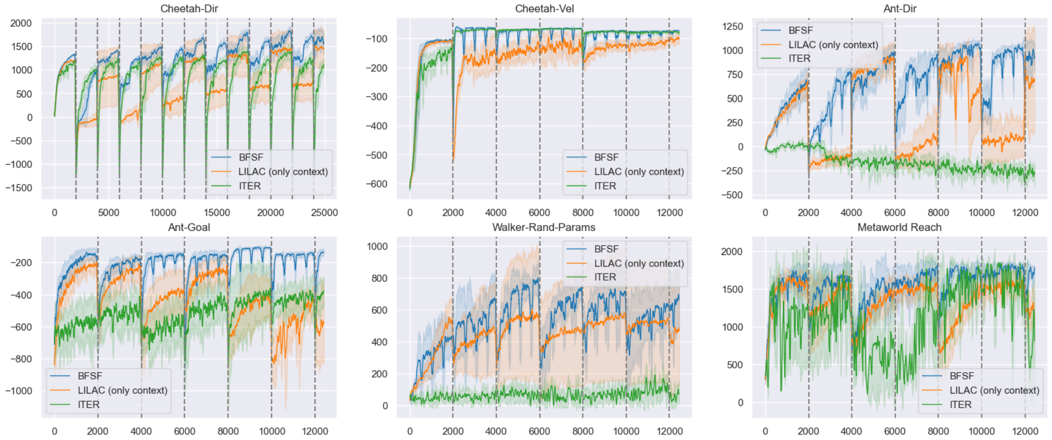


Figure 2: The learning curve of BFSF and other baselines, on the non-stationary MDPs based on 5 MuJoCo locomotion environments and 1 Meta-World environment. For clarity, we only display the curves for BFSF, LILAC, and ITER. Additional curves for CEMRL and CoMPs, along with implementation details, can be found in Appendix B.

	CHEETAH-DIR	CHEETAH-VEL	ANT-DIR	ANT-GOAL
BFSF	$\mathbf{1209.0 \pm 24.0}$	$\mathbf{-99.1 \pm 2.8}$	$\mathbf{671.0 \pm 37.7}$	$\mathbf{-204.9 \pm 9.3}$
LILAC	757.7 ± 94.9	-154.6 ± 2.5	318.5 ± 66.2	-426.6 ± 4.6
ITER	813.4 ± 19.0	-102.0 ± 2.8	-156.8 ± 25.7	-519.2 ± 50.0
CEMRL	852.6 ± 36.1	-196.0 ± 5.2	289.6 ± 18	-547.1 ± 67.6
COMPS	277.3 ± 40.2	-117.8 ± 6.1	-142.1 ± 6.7	-602.0 ± 72.2

	WALKER	REACH	ANT-DIR-INF	ANT-CIR-INF
BFSF	530.9 ± 59.8	$\mathbf{1543.4 \pm 93.5}$	$\mathbf{153.3 \pm 4.2}$	$\mathbf{165.2 \pm 6.8}$
LILAC	448.8 ± 279.1	1341.0 ± 47.7	85.5 ± 18.7	69.5 ± 1.6
ITER	65.6 ± 3.8	1112.1 ± 168.5	-338.5 ± 64.1	-337.0 ± 69.8
CEMRL	$\mathbf{577.9 \pm 68.1}$	977.5 ± 514.3	-11.8 ± 17.4	28.0 ± 13.7
COMPS	80.0 ± 34.4	526.9 ± 380.0	-151.0 ± 19.4	-121.1 ± 4.8

Table 1: The average return throughout the training process, comparing all the baselines. ‘Walker’ and ‘Reach’ are abbreviations for Walker-Rand-Params and Meta-World Reach, respectively.

of both BFSF and the baselines. Finally, for the third question, we conduct detailed ablation studies on the modules within BFSF, offering evidence of their effectiveness.

3.1 Mains Results

We evaluate the Bayesian Fast-Slow Framework (BFSF) on five MuJoCo environments and one Meta-World environment. The MuJoCo environments [28] focus on robotic locomotion and are based on the MuJoCo simulator, while Meta-World [35] is a benchmark designed for Multi-Task and meta-RL, specifically with robot manipulation tasks. These environments require adaptation across different reward functions (e.g., walking direction for Cheetah-Dir and Ant-Dir, target velocity for Cheetah-Vel, and goal location for Ant-Goal and Meta-World Reach), or across different dynamics (e.g., environment parameters for Walker-Rand-Params). These meta-RL environments are widely used in the meta-RL literature and are well-suited for non-stationary MDPs as well. We set the switching frequency of the underlying task to 2000 episodes.

We compare BFSF with four reproduced baselines. LILAC [31] is an algorithm for non-stationary MDPs that uses meta-RL techniques. It learns a latent variable to discriminate between tasks based on experiences. ITER [14] proposes an iterative approach to relearn the neural network, aiming to overcome non-stationarity. CEMRL [5] learns a task encoder from the gradients of a decoder and provides the task encoding to downstream RL. CoMPS [4] continuously alternates between two subroutines: learning a new task using RL and performing completely offline meta-learning to prepare for subsequent task learning.

As shown in Figure 2, BFSF outperforms the baselines in non-stationary environments. The task switching is indicated by the gray dashed lines for clarity. While we conduct experiments with a fixed switching interval, it is important to note that our algorithms are designed for the general setting where the task distribution and switching timing are completely unknown to the agent. For comprehensiveness, we also experiment with an unfixed switching interval in Section 3.3.

In general, all algorithms show gradual performance improvements within a single task phase but experience a sudden performance drop immediately after task switches. This decay is expected, as the new task is unfamiliar to the agent. However, we observe that the learning curve in the second phase does not increase as quickly as in the first, which we refer to as the interference phenomenon.

BFSF effectively mitigates interference, leading to better overall performance across continual phases. LILAC, based on meta-RL methods, provides good cross-task generalization. However, it fails to address interference, resulting in slower learning during the second task. ITER's iterative relearning approach is a solid defense against interference, but relying solely on this approach leads to the learning of elementary policies, which hinders further generalization and improvement. For clarity, we only compare the curves of two baselines in the given figure, while a full comparison of average return is provided in Table 1. Full experiment results on the learning curve is provided in Appendix B.2.

3.2 Infinite-World Simulation

While the main results in Section 3.1 demonstrate the superiority of BFSF in mitigating interference and achieving cross-task generalization, it remains unclear how such methods perform in more realistic scenarios. In the real world, there are typically no explicit task boundaries, no fixed task initializations, and no finite set of predefined tasks. To better approximate these characteristics, we introduce an Infinite-World Simulation in MuJoCo, an environment where the agent operates in a non-episodic, continuous manner without resets.

Unlike traditional MuJoCo benchmarks, where each episode ends and resets after a fixed number of steps (e.g., every 1000 steps), our infinite-world environment allows the agent to move seamlessly through a boundless plane without ever being reinitialized. This design leads to a non-episodic interaction flow, closely mimicking the persistent nature of real-world settings. To support this, we implement a dynamic terrain loading module that handles environment generation on the fly, avoiding memory overload while preserving the illusion of an endless space.

We design two signature environments to evaluate BFSF and baselines under this setup. Ant-Dir-Inf is a non-episodic, infinite-world variant of the standard Ant-Direction environment, where the agent is required to walk alternately left and right across the plane; each time a directional goal is reached, the target direction flips. Ant-Goal-Inf is derived from Ant-Goal, where the target moves along a circular trajectory of infinite radius, requiring the agent to constantly adjust and track it over time.

The corresponding results are reported in Table 1. In addition, Figure 3 visualizes the trajectories of different methods. BFSF consistently follow the evolving goal direction, while baseline methods react more slowly. This highlights BFSF's effectiveness in non-episodic and task-free environments.

3.3 Ablation Studies

The ablation studies on each module of BFSF are conducted to answer the question: How do the individual components of BFSF contribute to improving performance? The results show that the 'fast' policy, as a whole, alleviates interference and enhances overall performance in non-stationary MDPs. Additionally, the dual-reset mechanism and relabeling further support the 'fast' policy by enabling more effective learning.

Unfixed Switching Interval We conducted experiments with an unfixed switching interval, as shown in Figure 4(a). The overall performance exhibits a pattern similar to that observed in the main experiments with a fixed switching interval: the interference issue is mitigated by BFSF, and the agent's performance continues to improve as training progresses.

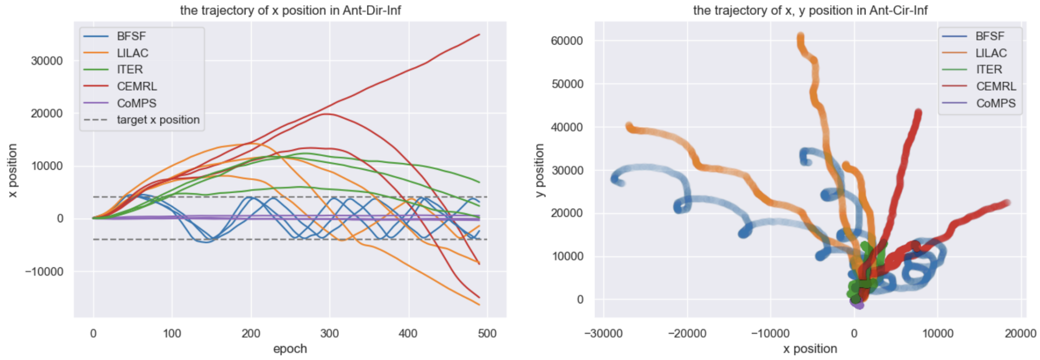
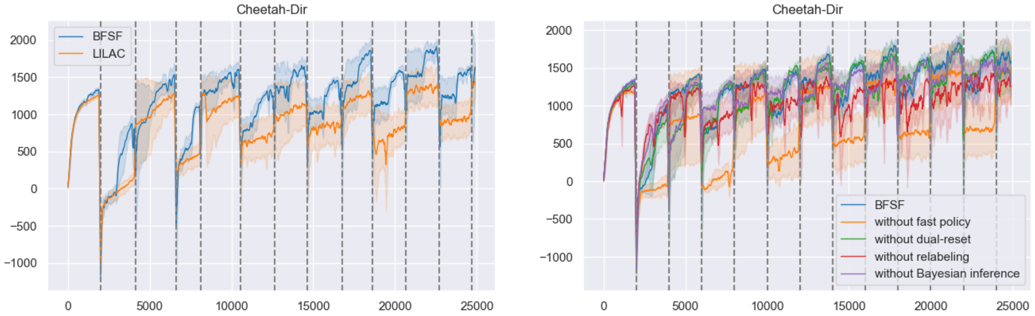


Figure 3: Visualized trajectories for Ant-Dir-Inf and Ant-Cir-Inf are shown. In Ant-Dir-Inf (left), only the BFSF algorithm successfully adapts quickly to the alternating goals between left and right directions. In Ant-Cir-Inf (right), only BFSF demonstrates rapid adaptation to the continuously moving goal along a circular path.



(a) BFSF with an unfixed switching interval.

(b) The ablation study of BFSF.

Figure 4: Ablation studies about the unfixed switching interval and other modules.

’Fast’ Policy Ablation The presence of the ‘fast’ policy, which learns from recent data, enables the agent to better adapt to changes in non-stationary MDPs, as shown in Figure 4(b). The curve labeled ‘without fast policy’ is identical to the baseline LILAC, as introduced in Section 3.1. While LILAC can generalize across tasks, it struggles to efficiently learn the optimal policy in the second task due to the interference problem. In contrast, BFSF addresses this issue, learning the second task at nearly the same speed as the first task, effectively overcoming interference. This trend is also observed in ongoing tasks, where interference does not negatively impact the performance of BFSF.

Dual-Reset Ablation The dual-reset mechanism ensures the effectiveness of the ‘fast’ policy. As shown in Figure 4(b), without the dual-reset, the learning curve exhibits lower performance due to a suboptimal ‘fast’ policy.

Relabeling Ablation As seen in Figure 4(b), relabeling significantly improves the learning efficiency of the ‘fast’ policy. With relabeling, the ‘fast’ policy can continuously improve its performance, even on later tasks. In contrast, BFSF without relabeling, as shown in Figure 5, struggles to achieve similar improvement in later tasks.

Bayesian Inference Ablation As introduced in Section 2.1, Bayesian inference provides a suitable estimate of the expected return for both the ‘fast’ and ‘slow’ policies. Without it, the estimation becomes less effective, and proper hyperparameter tuning may be required. As illustrated in Figure 4(b), the performance and resistance to interference deteriorate in the absence of Bayesian inference.

4 Related works

Non-Stationary MDPs Research on non-stationary MDPs primarily focuses on the challenge of recognizing potential tasks, as understanding the task transforms the non-stationary MDP into a fixed MDP. LILAC [31] first employs latent variable models to learn environment representations based on

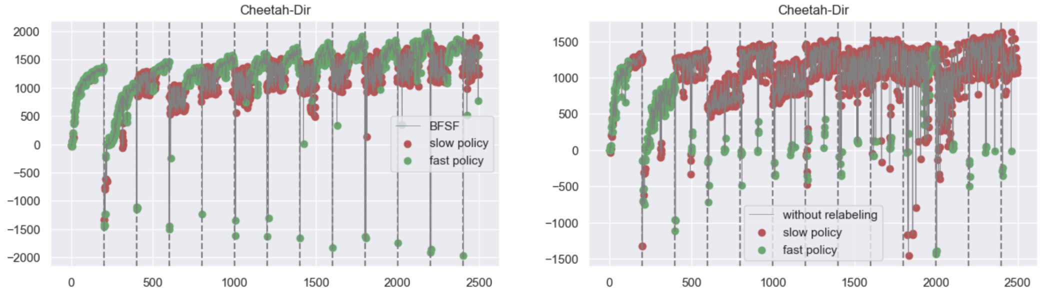


Figure 5: A comparison illustrating the performance of the 'slow' and 'fast' policies. The main difference is that with relabeling, the performance of the 'fast' policy remains higher throughout the phases, rather than significantly dropping after the initial phases.

current and past experiences, drawing inspiration from online learning and probabilistic inference. Subsequent works identified shortcomings in LILAC and proposed solutions. For instance, FANS-RL [10] models non-stationarity in terms of individual latent change factors and causal graphs. ITER [14] highlights the impact of non-stationarity on latent representations, a form of interference similar to the one discussed in our work, leading to the proposal of Iterated Relearning (ITER). Additionally, several theoretical works have also focused on non-stationary MDPs [1, 11, 2, 7]. However, none of these prior works simultaneously address both cross-task generalization and the interference problem.

Besides, continual RL shares similarities with non-stationary MDPs but focuses on different challenges, particularly catastrophic forgetting [25]. Although non-stationary MDPs and continual RL are often treated as distinct problems, their focus differs, primarily due to the assumption that task boundaries are known to agents in continual RL. As a result, research in continual RL focuses on designing submodules within the overall algorithm [30], such as replay buffers [6], network architecture [23], representation [22], or optimization strategies [21], rather than addressing the non-stationarity itself.

Meta-RL Meta-RL aims to enable agents to adapt more quickly to new tasks by leveraging prior experience from multiple tasks. It bears strong resemblance to non-stationary MDPs, but assumes that agents have full access to all tasks, thus ignoring interference effects. RL^2 [9] learns an agent's learning algorithm, enabling it to adapt quickly to new tasks by adjusting its internal update rule based on prior experience. MAML [12], a gradient-based meta-RL algorithm, seeks a set of model parameters that can be quickly adapted to new tasks with minimal gradient updates. In contrast, context-based meta-RL methods, such as PEARL [24] and VariBAD [37], leverage contextual information to enable more efficient adaptation. While traditional meta-RL assumes access to all tasks during training, recent research has explored meta-learning in the continual task setting [4, 5], which is closely related to our work on non-stationary MDPs.

Inference-related Works The issue of interference has been widely explored in related areas. In multi-task RL, task interference has been observed, and specialized network architectures have been proposed to mitigate this challenge [17, 8]. However, the source of interference in these works differs from that in non-stationary MDPs, where interference arises from unknown, streaming tasks. In continual RL, which focuses on maintaining high performance across incremental tasks, interference is also recognized and investigated at the representation level [18].

5 Conclusion

Non-stationarity poses a significant challenge when deploying RL agents in real-world environments. In addition to cross-task generalization through context-based algorithms, a challenge that has been thoroughly explored in previous works, we have identified that interference can severely hinder performance, especially when tasks conflict with each other. To address this issue, we introduce the Bayesian Fast-Slow Framework (BFSF), which incorporates a 'fast' policy that learns from recent history to prevent interference from previous tasks, and a 'slow' policy that maintains strong cross-task generalization. The use of Bayesian estimation ensures an effective and unbiased selection between the fast and slow policies, enhancing the framework's adaptability and robustness. We also introduce a dual-reset mechanism and data relabeling to further enhance efficiency. Experimental results demonstrate BFSF's effectiveness in resisting interference and show that it outperforms baseline methods in various non-stationary environments.

Although BFSF improves adaptability and efficiency in non-stationary MDPs, the current experiments are limited to a small set of environments. Other types of non-stationarity, such as blurred boundaries or stochastic non-stationary MDPs, have not yet been tested. Additionally, more realistic scenarios are needed to assess its applicability in real-world situations. In future work, we aim to develop more realistic benchmarks that align closely with real-world applications of non-stationary MDPs, while further testing BFSF and exploring new challenges.

6 Acknowledgment

This work was supported by AF Office of Scientific Research (AFOSR) under grant FA9550-25-1-0318 and with the generous support of the Amazon Research Award program.

References

[1] Alekh Agarwal, Sham Kakade, Mikael Henaff, and Wen Sun. Pc-pg: policy cover directed exploration for provable policy gradient learning. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA, 2020. Curran Associates Inc.

[2] Reda Alami, Mohammed Mahfoud, and Eric Moulines. Restarted bayesian online change-point detection for non-stationary markov decision processes. In Sarath Chandar, Razvan Pascanu, Hanie Sedghi, and Doina Precup, editors, *Proceedings of The 2nd Conference on Lifelong Learning Agents*, volume 232 of *Proceedings of Machine Learning Research*, pages 715–744. PMLR, 22–25 Aug 2023.

[3] Andrew Ang and Allan Timmermann. Regime changes and financial markets. *Annual Review of Financial Economics*, 4(Volume 4, 2012):313–337, 2012.

[4] Glen Berseth, Zhiwei Zhang, Grace Zhang, Chelsea Finn, and Sergey Levine. CoMPS: Continual meta policy search. In *Deep RL Workshop NeurIPS 2021*, 2021.

[5] Zhenshan Bing, David Lerch, Kai Huang, and Alois Knoll. Meta-reinforcement learning in non-stationary and dynamic environments. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3476–3491, 2023.

[6] Lucas Caccia, Rahaf Aljundi, Nader Asadi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. New insights on reducing abrupt representation change in online continual learning. *ArXiv*, abs/2203.03798, 2021.

[7] Wang Chi Cheung, David Simchi-Levi, and Ruihao Zhu. Reinforcement learning for non-stationary Markov decision processes: The blessing of (More) optimism. In Hal Daumé III and Aarti Singh, editors, *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pages 1843–1854. PMLR, 13–18 Jul 2020.

[8] Chuntao Ding, Zhichao Lu, Shangguang Wang, Ran Cheng, and Vishnu Naresh Boddeti. Mitigating task interference in multi-task learning via explicit task routing with non-learnable primitives. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7756–7765, June 2023.

[9] Yan Duan, John Schulman, Xi Chen, Peter L. Bartlett, Ilya Sutskever, and P. Abbeel. Rl^2: Fast reinforcement learning via slow reinforcement learning. *ArXiv*, abs/1611.02779, 2016.

[10] Fan Feng, Biwei Huang, Kun Zhang, and Sara Magliacane. Factored adaptation for non-stationary reinforcement learning. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 31957–31971. Curran Associates, Inc., 2022.

[11] Songtao Feng, Ming Yin, Ruiquan Huang, Yu-Xiang Wang, Jing Yang, and Yingbin Liang. Non-stationary reinforcement learning under general function approximation. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023.

[12] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning - Volume 70*, ICML'17, pages 1126–1135. JMLR.org, 2017.

[13] Weiyu Guo and Mark E. Wohar. Identifying regime changes in market volatility. *Journal of Financial Research*, 29(1):79–93, 2006.

[14] Maximilian Igl, Gregory Farquhar, Jelena Luketina, Wendelin Boehmer, and Shimon Whiteson. Transient non-stationarity and generalisation in deep reinforcement learning. In *International Conference on Learning Representations*, 2021.

[15] Michael Janner, Justin Fu, Marvin Zhang, and Sergey Levine. *When to trust your model: model-based policy optimization*. Curran Associates Inc., Red Hook, NY, USA, 2019.

[16] Leslie P Kaelbling, Michael L. Littman, and Anthony R. Cassandra. Planning and acting in partially observable stochastic domains. Technical report, USA, 1996.

[17] Menelaos Kanakis, David Bruggemann, Suman Saha, Stamatios Georgoulis, Anton Obukhov, and Luc Van Gool. Reparameterizing convolutions for incremental multi-task learning without task interference. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 689–707. Springer, 2020.

[18] Samuel Kessler, Jack Parker-Holder, Philip Ball, Stefan Zohren, and Stephen J. Roberts. Same state, different task: Continual reinforcement learning without interference. 36:7143–7151, Jun. 2022.

[19] Khimya Khetarpal, Matthew Riemer, Irina Rish, and Doina Precup. Towards continual reinforcement learning: A review and perspectives. *J. Artif. Intell. Res.*, 75:1401–1476, 2020.

[20] Vincent Liu, Han Wang, Ruo Yu Tao, Khurram Javed, Adam White, and Martha White. Measuring and mitigating interference in reinforcement learning. In Sarath Chandar, Razvan Pascanu, Hanie Sedghi, and Doina Precup, editors, *Proceedings of The 2nd Conference on Lifelong Learning Agents*, volume 232 of *Proceedings of Machine Learning Research*, pages 781–795. PMLR, 22–25 Aug 2023.

[21] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6470–6479, Red Hook, NY, USA, 2017. Curran Associates Inc.

[22] Divyam Madaan, Jaehong Yoon, Yuanchun Li, Yunxin Liu, and Sung Ju Hwang. Representational continuity for unsupervised continual learning. In *International Conference on Learning Representations*, 2022.

[23] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In Vittorio Ferrari, Martial Hebert, Cristian Sminchisescu, and Yair Weiss, editors, *Computer Vision – ECCV 2018*, pages 72–88, Cham, 2018. Springer International Publishing.

[24] Kate Rakelly, Aurick Zhou, Chelsea Finn, Sergey Levine, and Deirdre Quillen. Efficient off-policy meta-reinforcement learning via probabilistic context variables. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5331–5340. PMLR, 09–15 Jun 2019.

[25] ANTHONY ROBINS. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2):123–146, 1995.

[26] Amila Thibbotuwawa, Grzegorz Bocewicz, Grzegorz Radzki, Peter Nielsen, and Zbigniew Banaszak. Uav mission planning resistant to weather uncertainty. *Sensors*, 20(2), 2020.

[27] Amila Thibbotuwawa, Grzegorz Bocewicz, Banaszak Zbigniew, and Peter Nielsen. A solution approach for uav fleet mission planning in changing weather conditions. *Applied Sciences*, 9(19), 2019.

[28] Emanuel Todorov, Tom Erez, and Yuval Tassa. Mujoco: A physics engine for model-based control. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 5026–5033, 2012.

[29] Michael Wan, Jian Peng, and Tanmay Gangwani. Hindsight foresight relabeling for meta-reinforcement learning. In *International Conference on Learning Representations*, 2022.

[30] Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 46(8):5362–5383, 2024.

[31] Annie Xie, James Harrison, and Chelsea Finn. Deep reinforcement learning amidst continual structured non-stationarity. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 11393–11403. PMLR, 18–24 Jul 2021.

[32] Huazhe Xu, Yuanzhi Li, Yuandong Tian, Trevor Darrell, and Tengyu Ma. Algorithmic framework for model-based reinforcement learning with theoretical guarantees. *CoRR*, abs/1807.03858, 2018.

[33] Tengye Xu, Zihao Li, and Qinyuan Ren. Meta-reinforcement learning robust to distributional shift via performing lifelong in-context learning. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

[34] Zhuoran Yang, Yuchen Xie, and Zhaoran Wang. A theoretical analysis of deep q-learning, 2020.

[35] Tianhe Yu, Deirdre Quillen, Zhanpeng He, Ryan Julian, Karol Hausman, Chelsea Finn, and Sergey Levine. Meta-world: A benchmark and evaluation for multi-task and meta reinforcement learning. In *Conference on Robot Learning (CoRL)*, 2019.

[36] Tianhe Yu, Garrett Thomas, Lantao Yu, Stefano Ermon, James Zou, Sergey Levine, Chelsea Finn, and Tengyu Ma. Mopo: Model-based offline policy optimization, 2020.

[37] Luisa Zintgraf, Sebastian Schulze, Cong Lu, Leo Feng, Maximilian Igl, Kyriacos Shiarlis, Yarin Gal, Katja Hofmann, and Shimon Whiteson. Varibad: variational bayes-adaptive deep rl via meta-learning. *J. Mach. Learn. Res.*, 22(1), January 2021.

NeurIPS Paper Checklist

1. **Claims**

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

   Answer: [Yes]

   Justification: The introduction mentions all the parts in the text.

   Guidelines:

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

2. **Limitations**

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

   Answer: [Yes]

   Justification: The limitations are included in the conclusion.

   Guidelines:

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

3. **Theory assumptions and proofs**

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

   Answer: [Yes]

Justification: The assumptions and proof are in the main text and Appendix A.1.

Guidelines:

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

4. **Experimental result reproducibility**

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

   Answer: [Yes]

   Justification: The implementation details can be found in the code.

   Guidelines:

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

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

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

Answer: [Yes]

Justification: The code is in the supplementary materials.

Guidelines:

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

6. **Experimental setting/details**

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

   Answer: [Yes]

   Justification: The implementation details can be found in the code.

   Guidelines:

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

7. **Experiment statistical significance**

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

   Answer: [Yes]

   Justification: The error bars are provided in the Table 1.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
   - The assumptions made should be given (e.g., Normally distributed errors).
   - It should be clear whether the error bar is the standard deviation or the standard error of the mean.

- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

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

   Answer: [Yes]

   Justification: The implementation details can be found in the code.

   Guidelines:

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

9. **Code of ethics**

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

   Answer: [Yes]

   Justification: The research satisfies the NeurIPS Code of Ethics.

   Guidelines:

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

10. **Broader impacts**

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

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to

generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.

- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

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

Answer: [NA]

Justification:

Guidelines:

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

12. **Licenses for existing assets**

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

Answer: [Yes]

Justification: The creator of the code is mentioned in the code.

Guidelines:

- The answer NA means that the paper does not use existing assets.
- The authors should cite the original paper that produced the code package or dataset.
- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

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

Answer: [NA]

Justification:

Guidelines:

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

14. **Crowdsourcing and research with human subjects**

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

    Answer: [NA]

    Justification:

    Guidelines:

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

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

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

    Answer: [NA]

    Justification:

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

Justification:

Guidelines:

- The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy (`https://neurips.cc/Conferences/2025/LLM`) for what should or should not be described.

Technical Appendices and Supplementary Material

A About Theorem 2.1

A.1 Proof of Theorem 2.1

Proof. First, since Bayesian estimation provides an unbiased estimate, the suboptimality of BFSF is bounded by the better performance between the fast and slow policies.

$$\text{Suboptimality}(BFSF) \leq \min(\text{Suboptimality}(Fast), \text{Suboptimality}(Slow)) \tag{4}$$

We analyze the suboptimalimality of the 'fast' policy $|\eta_M(\pi^*_M) - \eta_M(\pi^*_{M'})|$, in the following proof.

Let MDP M have the dynamic function p and reward function r. Similarly, let MDP M' have the dynamic function p' and reward function r'. We denote $M_{p,r}$ as the MDP with dynamics function p and reward function r. In Equation 5, we decompose $|\eta_M(\pi^*_M) - \eta_M(\pi^*_{M'})|$ and use Theorem A.1, Lemma A.2, A.3 to complete the proof.

$$\begin{aligned}
&|\eta_M(\pi^*_M) - \eta_M(\pi^*_{M'})| \\
\leq&|\eta_M(\pi^*_M) - \eta_{M'}(\pi^*_{M'})| + |\eta_{M'}(\pi^*_{M'}) - \eta_M(\pi^*_{M'})| \\
=&|\eta_{M_{p,r}}(\pi^*_{M_{p,r}}) - \eta_{M_{p',r'}}(\pi^*_{M_{p',r'}})| + |\eta_{M_{p',r'}}(\pi^*_{M_{p',r'}}) - \eta_{M_{p,r}}(\pi^*_{M_{p',r'}})| \\
\leq&|\eta_{M_{p,r}}(\pi^*_{M_{p,r}}) - \eta_{M_{p',r'}}(\pi^*_{M_{p',r'}})| \\
&+ |\eta_{M_{p',r'}}(\pi^*_{M_{p',r'}}) - \eta_{M_{p,r'}}(\pi^*_{M_{p',r'}})| + |\eta_{M_{p,r'}}(\pi^*_{M_{p',r'}}) - \eta_{M_{p,r}}(\pi^*_{M_{p',r'}})| \\
\leq&[\mathcal{D}_{\ell_1}(p_{M_1}, p_{M_2})(r_{max} + V_{max}) + r_{diff}]H + |U_{r_1,r_2}(\pi^*_{M_2})| + \frac{1}{2}V_{max}\mathcal{D}_{\ell_1}(p_{M_2}(s,a), p_{M_1}(s,a))
\end{aligned} \tag{5}$$

□

Theorem A.1. *(Relabeling gap 1)Let M_1, M_2 be two finite-horizon MDPs with the same reward function r. Then the distance of $\eta_{M_1}(\pi^*_{M_1})$ and $\eta_{M_2}(\pi^*_{M_2})$ is bounded by*

$$|\eta_{M_1}(\pi^*_{M_1}) - \eta_{M_2}(\pi^*_{M_2})| \leq \epsilon_h, \tag{6}$$

where $\epsilon_h = [\mathcal{D}_{\ell_1}(p_{M_1}, p_{M_2})(r_{max} + V_{max}) + r_{diff}](H - h)$, $\forall h \in [H], s \in \mathcal{S}$.

Proof. Since $\pi^*_{M_1}$ is the optimal policy of MDP M_1 and $\pi^*_{M_2}$ is the optimal policy of MDP M_2, $\eta_{M_1}(\pi^*_{M_1}) = V^*_{M_1}, \eta_{M_2}(\pi^*_{M_2}) = V^*_{M_2}$.

We begin our proof from the final horizon, $h = H$, and use the closeness at horizon h to establish the closeness at horizon $h - 1$.

For the final horizon $h = H$, we have $V^*_{M,H}(s) = 0 \ \forall M$ since it is the terminal state. Therefore, $||V^*_{M_1,H}(s) - V^*_{M_2,H}||_\infty \leq 0 = \epsilon_H$. Suppose

$$\forall s \in \mathcal{S}_h, ||V^*_{M_1,h}(s) - V^*_{M_2,h}||_\infty \leq \epsilon_h. \tag{7}$$

We need to prove

$$\forall s \in \mathcal{S}_{h-1}, ||V^*_{M_1,h-1}(s) - V^*_{M_2,h-1}||_\infty \leq \epsilon_{h-1}. \tag{8}$$

It is equivalent to prove

$$-\epsilon_{h-1} \leq V^*_{M_2,h-1}(s) - V^*_{M_1,h-1}(s) \leq \epsilon_{h-1}. \tag{9}$$

For simplicity, but without loss of generality, we will prove the inequality on the right-hand side of the above equation.

$$
\begin{aligned}
LHS =&\max_{a\in\mathcal{A}}\{\sum_{s'\in\mathcal{S}_h} p_{M_2}(s'|s,a)(r_2(s,a)+\gamma V^*_{M_2,h}(s'))\}\\
&-\max_{a\in\mathcal{A}}\{\sum_{s'\in\mathcal{S}_h} p_{M_1}(s'|s,a)(r_1(s,a)+\gamma V^*_{M_1,h}(s'))\}\\
=&\max_{a\in\mathcal{A}}\{\sum_{s'\in\mathcal{S}_h} [p_{M_1}(s'|s,a)(r(s,a)+\gamma V^*_{M_1,h}(s'))\\
&+(p_{M_2}(s'|s,a)r_2(s,a)-p_{M_1}(s'|s,a)r_1(s,a))+p_{M_2}(s'|s,a)V^*_{M_2,h}(s')\\
&-p_{M_1}(s'|s,a)V^*_{M_1,h}(s')]\}-\max_{a\in\mathcal{A}}\{\sum_{s'\in\mathcal{S}_h} p_{M_1}(s'|s,a)(r(s,a)+\gamma V^*_{M_1,h}(s'))\}\\
\le&\max_{a\in\mathcal{A}}\{\sum_{s'\in\mathcal{S}_h} p_{M_1}(s'|s,a)(r(s,a)+\gamma V^*_{M_1,h}(s'))\}\\
&+\max_{a\in\mathcal{A}}\{\sum_{s'\in\mathcal{S}_h} [(p_{M_2}(s'|s,a)r_2(s,a)-p_{M_1}(s'|s,a)r_1(s,a))\\
&+p_{M_2}(s'|s,a)V^*_{M_2,h}(s')-p_{M_1}(s'|s,a)V^*_{M_1,h}(s')]\}\\
&-\max_{a\in\mathcal{A}}\{\sum_{s'\in\mathcal{S}_h} p_{M_1}(s'|s,a)(r(s,a)+\gamma V^*_{M_1,h}(s'))\}\\
=&\max_{a\in\mathcal{A}}\{\sum_{s'\in\mathcal{S}_h} [(p_{M_2}(s'|s,a)r_2(s,a)-p_{M_1}(s'|s,a)r_1(s,a))\\
&+p_{M_2}(s'|s,a)V^*_{M_2,h}(s')-p_{M_1}(s'|s,a)V^*_{M_1,h}(s')]\}\\
=&\max_{a\in\mathcal{A}}\{\sum_{s'\in\mathcal{S}_h} [(p_{M_2}(s'|s,a)-p_{M_1}(s'|s,a))r_2(s,a)+\sum_{s'\in\mathcal{S}_h} p_{M_1}(s'|s,a)(r_2(s,a)-r_1(s,a))\\
&+p_{M_2}(s'|s,a)V^*_{M_2,h}(s')-p_{M_1}(s'|s,a)V^*_{M_1,h}(s')]\}\\
\le&\max_{a\in\mathcal{A}}\{\sum_{s'\in\mathcal{S}_h} |p_{M_2}(s'|s,a)-p_{M_1}(s'|s,a)|\}r_{max}+\max_{a\in\mathcal{A}}\{\sum_{s'\in\mathcal{S}_h} p_{M_1}(s'|s,a)r_{diff}\}\\
&+\max_{a\in\mathcal{A}}\{\sum_{s'\in\mathcal{S}_h} (p_{M_2}(s'|s,a)-p_{M_1}(s'|s,a))V^*_{M_2,h}(s')+p_{M_1}(s'|s,a)(V^*_{M_2,h}(s')-V^*_{M_1,h}(s'))\}\\
\le&\{D_{\ell_1}(p_{M_1}(\cdot|s,a),p_{M_2}(\cdot|s,a))\}(r_{max}+V_{max})+r_{diff}+1\cdot(V^*_{M_2,h}(s')-V^*_{M_1,h}(s'))\\
\le&[D_{\ell_1}(p_{M_1},p_{M_2})(r_{max}+V_{max})+r_{diff}]+1\cdot\epsilon(h)\\
=&[D_{\ell_1}(p_{M_1},p_{M_2})(r_{max}+V_{max})+D_{\ell_1}(p_{M_1},p_{M_2})(r_{max}+V_{max})+r_{diff}](H-h)\\
=&[D_{\ell_1}(p_{M_1},p_{M_2})(r_{max}+V_{max})+r_{diff}](H-h+1).
\end{aligned}
\tag{10}
$$

□

Lemma A.2. *Let M_1, M_2 be two MDPs with the same dynamics function, but different reward functions r_1, r_2. Define $U_{r_1,r_2}(\pi) = \mathbb{E}_{(s,a)\sim\rho^{\pi}_{M_1}}[r_2(s,a) - r_1(s,a)]$, which characterizes how erroneous the model is along trajectories induced by π. Then*

$$\eta_{M_2}(\pi) - \eta_{M_1}(\pi) = U_{r_1,r_2}(\pi) \tag{11}$$

Proof. We know that $\tilde{M}$ and $\hat{M}$ shares the same transition dynamics p, but different reward functions $\tilde{r}(s,a) = \hat{r}(s,a) - \lambda u(s,a)$. Therefore,

$$
\begin{aligned}
\eta_{M_2}(\pi) =&\mathbb{E}_{(s,a)\sim\rho^{\pi}_{M_1}}[r_2(s,a)]\\
=&\mathbb{E}_{(s,a)\sim\rho^{\pi}_{M_1}}[r_1(s,a)+(r_2(s,a)-r_1(s,a))]\\
=&\mathbb{E}_{(s,a)\sim\rho^{\pi}_{M_1}}r_1(s,a)-\mathbb{E}_{(s,a)\sim\rho^{\pi}_{M_1}}(r_2(s,a)-r_1(s,a))\\
=&\eta_{M_1}(\pi)+U_{r_1,r_2}(\pi).
\end{aligned}
\tag{12}
$$

□

Lemma A.3. *(Telescoping lemma) [36, 32]. Let M_1 and M_2 be two MDPs with the same reward $r(s,a)$, but different dynamics p_{M_1} and p_{M_2} respectively. Let*

$$\begin{aligned} &{G_{M_2}}^{\pi}(s,a) := \\ &\qquad \mathbb{E}_{s' \sim p_{M_2}(s,a)}[V^{\pi}_{M_1}(s')] - \mathbb{E}_{s' \sim p_{M_1}(s,a)}[V^{\pi}_{M_1}(s')], \end{aligned} \tag{13}$$

Then,

$$\eta_{M_2}(\pi) - \eta_{M_1}(\pi) = \gamma \mathbb{E}_{(s,a)\sim \rho^{\pi}_{M_2}}[G^{\pi}_{M_2}(s,a)]. \tag{14}$$

For each $s \in \mathcal{S}, a \in \mathcal{A}$, a ℓ_1-based bound of $|G^{\pi}_{M_2}(s,a)|$ is

$$|G^{\pi}_{M_2}(s,a)| \leq \frac{1}{2} V_{max} \delta_{\ell_1}(p_{M_2}(s,a), p_{M_1}(s,a)). \tag{15}$$

A.2 Bound Considering the Optimization Error

Theorem 2.1 accounts for the error introduced by the relabeling process. To maintain consistency with prior work, we explicitly incorporate optimization suboptimality, following the approach in [34], by considering the policy obtained after K policy-update iterations $\pi^K_{M'}$, rather than the idealized optimal policy $\pi^*_{M'}$.

$$\begin{aligned} &\text{Total Suboptimality} \\ \leq &|\eta_M(\pi^*_M) - \eta_M(\pi^K_{M'})| \\ \leq &|\eta_M(\pi^*_M) - \eta_{M'}(\pi^*_{M'})| + |\eta_{M'}(\pi^*_{M'}) - \eta_{M'}(\pi^K_{M'})| + |\eta_{M'}(\pi^K_{M'}) - \eta_M(\pi^K_{M'})| \\ \leq &[D_1(p_M, p_{M'})(r_{\max} + V_{\max}) + r_{\text{diff}}]H + |U_{r,r'}(\pi^*_{M'})| + \frac{1}{2} V_{\max} D_1(p_{M'}(s,a), p_M(s,a)) \\ &+ C \cdot \frac{\phi_{\mu,\sigma} \cdot \gamma}{(1-\gamma)^2} \cdot |A| \cdot (\log n)^{1+2\xi^*} \cdot n^{(\alpha^*-1)/2} + \frac{4\gamma^{K+1}}{(1-\gamma)^2} \cdot R_{\max}, \end{aligned} \tag{16}$$

where each term explicitly captures different sources of error:

- The **first 3 terms** (matching our original Theorem 3.1) represent the suboptimality caused by the **relabeling process**, quantifying the error introduced due to differences in dynamics and reward functions between the original MDP M and the relabeled MDP M'.
- The newly introduced **4th and 5th terms** explicitly quantify the **optimization suboptimality**, representing errors arising from finite-sample approximations and iterative optimization.

B Experiments

B.1 Implementation Details

The experiments are repeated three times, with the mean and standard deviation shown in the curves and table.

The common hyperparameters are consistent with the original PEARL implementation. Additionally, the context consists of 200 episodes, and the relabeling percentage is set to 50% (i.e., half of the used batch is relabeled). The window of recent data $w = 100$ episodes. The reset frequency $\nu = 50$ episodes. The discount factor $\gamma = 0.99$.

B.2 Full Learning Curves

A complete comparison of the learning curves for all four baselines is provided in Figure 6.

B.3 Sensitive Studies and Other Experiments

Sensitive study on the relabeling percentage We performed an sensitive study on the relabeling percentage in the Table 2.

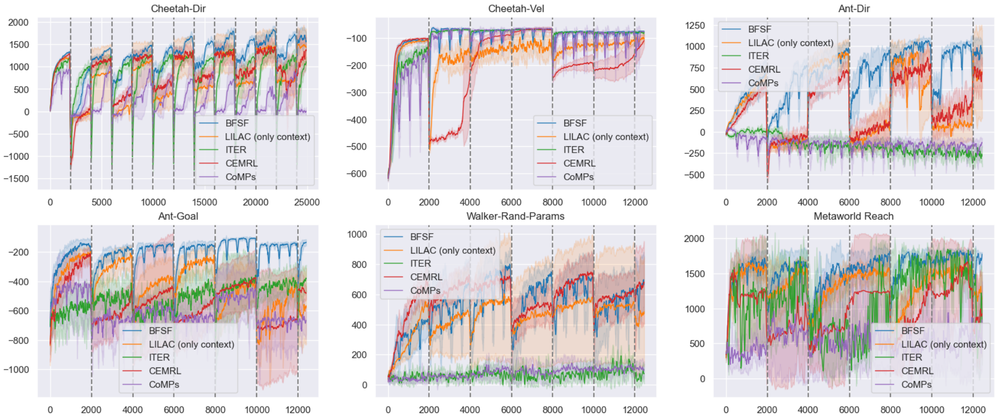


Figure 6: The learning curve of BFSF with all the baselines including LILAC, ITER, CEMRL and CoMPs.

Relabel Percentage (%)	0	10	20	30	40	50
Average Return	1055.5	954.9	1046.3	1097.5	1186.4	1039.4

Relabel Percentage (%)	60	70	80	90	100
Average Return	1217.7	1200.9	1148.1	1027.6	1129.1

Table 2: Sensitive study on the relabeling percentage

Sensitive study on the window of recent data w and the reset frequency ν is shown in Table 3. For reference, the average return of the baseline (LILAC) is 757.8. The results indicate that performance of our method is relatively insensitive to the choice of window size w and reset frequency ν.

The experiment in gradually changing environments To validate our method in dynamically changing environments, we conducted supplementary experiments in Table 4 in a 180-task gradually changing Ant-Goal environment, with goals evenly distributed around a circle.

The experiment in stochastically changing environments As to settings with highly stochastic task boundaries, we conduct experiments on a Cheetah-Dir environment in Table 5. This environment consists of two tasks: moving forward and moving backward. A boundary period (one-third of the task length) exists during task switching, with each task having a probability of 50%.

C Algorithm Details

C.1 Data Relabeling

To elaborate, relabeling is accomplished by a learned dynamics and reward model $s', r = f(s, a)$ [15], which estimates the next state s' and reward r after taking action a in the state s. We can

	$w = 10$	$w = 50$	$w = 100$	$w = 150$	$w = 200$
BFSF	1163.3	1273.6	1209.0	1121.1	945.5

	$\nu = 10$	$\nu = 25$	$\nu = 50$	$\nu = 75$	$\nu = 100$
BFSF	1250.6	1169.0	1209.0	1132.1	1116.6

Table 3: Sensitive study on the window of recent data w and the reset frequency ν.

	BFSF	Baseline: LILAC
Gradual Ant-Goal	-328.5 ± 2.1	-619.2 ± 3.7

Table 4: The experiment results in a gradual Ant-Goal environment. BFSF outperforms the baseline, showing its ability to adapt continuously to a changing environment.

	BFSF	Baseline: LILAC
Stochastic Ant-Goal	887.8 ± 21.6	615.4 ± 5.7

Table 5: The experiment results in a stochastic Ant-Goal environment. BFSF significantly outperforms baseline LILAC in such an environment.

substitute the original next state and reward in the experience replay, even from different tasks, with those predicted by the model, represented as $s'_{relabel}, r_{relabel} = f(s, a)$.

Relabeling is widely used in the RL community [29, 33]. The underlying principle is to maximize data reuse for sample efficiency. In our work, given that the environment evolves over time, we leverage a context-based dynamics model $f(s, a, c)$, which provides different dynamics depending on context c.

As discussed in [29, 33] and confirmed by our ablation studies, data relabeling substantially enhances sample efficiency. In our work, we adopt data relabeling to mitigate the issue of performance degradation in a non-stationary environment, because the increased amount of relabeled data allow the 'fast' policy to better and faster adapt to these tasks, which is verified by our ablation results.

C.2 Motivation for the Dual-Reset Mechanism

The dual-reset mechanism was introduced upon observing that RL algorithms tend to encounter performance degradation when alternating between tasks, a trend corroborated by prior research [4]. This mechanism effectively addresses the issue. We present the phase performance for SAC in the alternating Cheetah-Dir environment (where the two tasks are moving forward and backward) in Table 6.

C.3 Bayesian Fast-Slow Framework

We only utilize data within a recent window, which keeps the effective sample size small, and we set the prior value to a dynamically updated upper bound to encourage exploration. As a result, if a policy has not been sufficiently selected, the prior strongly influences the Bayesian estimate, leading to a large posterior value that naturally encourages exploration of that policy.

Phase performance	Task 1	Task 2	Task 1	Task 2	Task 1
Without dual-reset	954.6	117.8	92.0	8.6	-27.4
With dual-reset	759.7	625.6	697.5	409.8	782.7

Table 6: In the first phase (Task 1), SAC without the dual-reset mechanism performs well, even outpacing the SAC with dual-reset. However, during the alternating tasks, the performance degrades significantly.

D Posterior Calculation of Normal Distributions

Assume ϕ is known.

$$\begin{aligned}
p(\mu|\{R_{i_1}, R_{i_2}, \cdots\}, 1/\phi) &\propto p(\mu)p(\{R_{i_1}, R_{i_2}, \cdots\}|\mu, 1/\phi) \\
&\propto \exp\{-\frac{\phi_0}{2}(\mu-\mu_0)^2\} \times \exp\{-\frac{n\phi}{2}(\mu-\overline{y})^2\} \\
&\propto \exp\{-\frac{1}{2}(\phi_0+n\phi)\mu^2 + \frac{1}{2}(2\mu_0\phi_0 + 2n\phi\overline{y})\mu\} \\
&\propto \exp\{-\frac{1}{2}(\phi_0+n\phi)(\mu - \frac{\phi_0\mu_0+n\phi\overline{y}}{\phi_0+n\phi})^2\} \\
&\sim \text{Normal}(\mu_1, \sigma_1^2), \\
\text{where } \mu_1 = \frac{\phi_0\mu_0 + n\phi\overline{R}}{\phi_0+n\phi}&, \sigma_1^2 = \frac{1}{\phi_0+n\phi}.
\end{aligned} \tag{17}$$

E Visualization of Choosing Slow/Fast Policies

We presented visualization in left sub-figure of Figure 5 in our paper (attached here in Figure 7). In the left sub-figure, the 'fast' policy predominates in the selection during the initial phases, while the 'slow' policy shows competitive performance as the amount of accumulated data from different tasks increases. Additionally, in the latter part of a single phase, the 'fast' policy surpasses the 'slow' policy after learning from relabeled recent data.

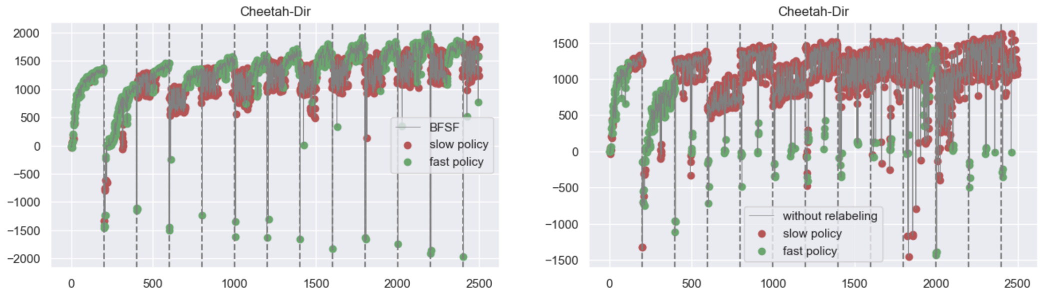


Figure 7: A comparison illustrating the performance of the 'slow' and 'fast' policies. The main difference is that with relabeling, the performance of the 'fast' policy remains higher throughout the phases, rather than significantly dropping after the initial phases.